# ToolBox Mag n° 2

Décembre 1990

## Sommaire de la revue :


### Editorial

| | | | |
|---|---|---|---|
| Pour le plaisir, sans sectarisme, avec confiance | E.Weyland | Page | 3 |

### Initiations

| | | | |
|---|---|---|---|
| Introduction au C sur GS : seconde partie | F. Uhrich | Page | 4 |
| Initiation aux Ressources du GS : première partie | B. Fournier | Page | 22 |

### Programmes

| | | | |
|---|---|---|---|
| GS Puzzle : un jeu de puzzle avec les images GS | B. Boissière | Page | 37 |
| PicMaster GS : convertissez les images GS | B. Fournier | Page | 26 |
| Un Tool pour jouer les musiques SoundSmith | O. Goguel | Page | 8 |

### News

| | | | |
|---|---|---|---|
| Octobre 1990 : Apple relance l'Apple II ! | J.Y. Bourdin | Page | 17 |
| Exotica : nouvelles du monde extérieur | F. Hermellin | Page | 35 |
| Made in France : les produits du terroir | B. Fournier | Page | 34 |
| GS News : les nouvelles du GS | E. Weyland | Page | 42 |

### Prêt à porter

| | | | |
|---|---|---|---|
| Revue hard : le Syquest, suite et fin | H. Loiseleux | Page | 18 |
| Revue soft : Block-Out, un Tetris en 3D | F. Hermellin | Page | 49 |
| Revue soft : Tarot, de F. Uhrich | L. Bourdin | Page | 25 |
| Supplique aux utilisateurs de GS | J.Y. Bourdin | Page | 34 |
| Fontes GS, suite | J.Y. Bourdin | Page | 33 |
| Bidouilles, patches, trucs | J.Y. Bourdin | Page | 41 |

### Infos / Dialogues

| | | | |
|---|---|---|---|
| Sommaire de la disquette | Toolbox-Mag | Page | 2 |
| Précisions sur Toolbox-Mag | J.Y. Bourdin | Page | 19 |
| Courrier des lecteurs | Vous... | Page | 48 |
| Quand Toolbox-Mag craque... | J.Y. Bourdin | Page | 50 |
| Abonnez vos amis | Toolbox-Mag | Page | 50-b |


Prix: 590F par abonnement.

*Note to our english-speaking readers: you will find a small text file for you, on the disk, in the ISUMMARIES folder.*

# Sommaire de la disquette Toolbox-Mag No 2 (🍎)

*Rappelons-le, la disquette est une partie du magazine. Elle ne peut pas tourner toute seule, et pour s'en servir, il faut lire la revue. Voici donc le catalogue de la partie disque du N°2:*

/ANIM
Dès son numéro 2, **Toolbox-Mag est en couleurs et en musique.** N'oubliez pas de booter la disquette. Merci aux auteurs! Le dossier /ANIM contient quatre fichiers nécessaires pour faire tourner l'animation de présentation. Ne touchez pas à ce dossier.

/ANNONCES
Nos petites annonces gratuites sur disque.

/C.FACILE
Contient le source de la commande Get pour APW publiée dans Toolbox-Mag N°1, le source d'une commande Purge, et d'autres programmes en C. Voyez l'article de F. Uhrich pour toutes précisions.

/DERNIERE.HEURE
Nous avons décidé de réserver quelques Ko pour des infos (ou des corrections éventuelles de dernière minute), pendant que la revue est chez l'imprimeur. Lisez les fichiers texte qui y sont présents.

/GS.PUZZLE
Le puzzle de B. Boissière, ainsi que son source (fichier texte et fichier ressources) en TML Pascal. Le fichier appelé 'Van.Gogh' est une image sadique pour le puzzle.

/ICONS
Des icônes pour le Finder.

/INDEX
Contient un fichier Base de Données Appleworks-GS appelé "Index.Modele". Ce fichier contiendra les références de toutes les questions abordées dans Toolbox-Mag dès que... vous les y aurez mises. Voyez l'encadré "Au boulot, lecteurs!".

/MJUKE.TML.PAS
Une Unit (source et objet) d'interface entre l'outil Music Studio de Stéphane Hadinger publié dans le numéro 1 et TML Pascal II. Merci à Mr Destelle.

/PICMASTER.GS
Contient le programme de conversion d'images et de ressources graphiques de B. Fournier. Le dossier /SOURCE comprend ses sources. Toutes explications sont à trouver dans l'article de B. Fournier.
A noter: faute de place, nous avons dû mettre les versions compilées des Units TML Pascal sur la disquette Bis. Il faudra donc recompiler les sources si vous voulez éditer le programme. Même chose pour les images d'exemples.

/SOUNDSMITH.TOOL
Contient le Tool d'O. Goguel pour jouer les musiques SoundSmith. Placez tout de suite le fichier "TOOL.219" dans le sous-catalogue /SYSTEM/TOOLS de votre disque système. Vous pouvez ensuite lancer le fichier appelé "ST.Exemple.Asm" pour avoir une démonstration de cet outil.
Le dossier /SOUNDSMITH.TOOL/MUSIC contient trois musiques de démonstration. Le fichier texte "User.Tool" contient une discussion pour savoir quel statut donner aux outils Toolbox-Mag.
Pour tous les autres fichiers de ce dossier, voir l'article d'O. Goguel dans ce numéro.

/SUMMARIES
Contient un petit fichier texte écrit dans un anglais fort approximatif pour nos lecteurs anglo-saxons.

FINDER.ROOT et PRODOS
Finder.Root permet d'afficher les icônes des applications. Prodos est un faux. Vous pouvez le jeter.


## Toolbox-Mag

**Les auteurs de Toolbox-Mag n° 2:**
H. Aalbers
O. Bailly-Maître
B. Boissière
J.Y. Bourdin
L. Bourdin
D. Delay
P. Desnoues
J. Destelle
B. Fournier
O. Goguel
F. Hermellin
Y. Kœnig
L. Picamelot
F. Uhrich
E. Weyland

**Directeur de la Publication**
Eric Weyland

**Rédacteur en Chef**
Jean-Yves Bourdin

**PAO - Maquettiste**
Dominique Digoy

**Secrétariat**
Janine Loiseleux

**Conseil de Rédaction**
Bernard Fournier
Olivier Goguel
Stéphan Hadinger
François Hermellin
Yvan Kœnig
Hubert Loiseleux
Claude Pélisson
Jean Luc Schmitt
François Uhrich

**Rédaction, Siège Social**
6, rue Henri-Barbusse
95100 Argenteuil
Téléphone: (1) 30 76 18 64
Télécopie: (1) 39 47 44 08

**Impression**
Imprimerie/Reprographie /Graphisme
Argenteuil

## Avertissements légaux

"GS" est un sigle déposé par les automobiles Citroën. La Pomme est un symbole déposé par Eve et le Serpent. Apple, Apple II GS, le logo Apple, Macintosh et le logo Macintosh, ImageWriter, LaserWriter et 'Fatal System Error 911' sont des marques déposées d'Apple Computer. AppleWorks-GS est une marque déposée par Claris-USA, à l'insu de Claris-France.

Toolbox-Mag est un magazine qui prend l'Apple II GS au sérieux. De ce fait même, il est indépendant d'Apple Computer et d'Apple Computer France, auxquels il n'est, grâce au ciel, pas rattaché.

Toolbox-Mag ne peut pas offrir une garantie supérieure à celle qu'offre le chapitre "Limitations de Garantie et de Responsabilité" des documentations officielles Apple, soit quasiment rien. Ecrit par des amateurs du GS, il s'engage néanmoins à publier des programmes ne contenant pas plus de bugs que GS Write, et à maintenir dans ses colonnes un standard de compétence au moins égal au niveau moyen des concessionnaires Apple français en matière de GS...

Toolbox-Mag est disponible uniquement par abonnement et par correspondance. Le magazine est composé inséparablement d'une revue sur papier et d'une disquette magnétique pour ordinateur Apple // GS. Certains avertissements légaux, modes d'emploi, etc, ou tous autres textes, peuvent être placés sur la disquette, et nécessiter un Apple II GS pour être lus: ils n'en sont pas moins écrits dans Toolbox-Mag, dont ils font partie intégrante.

Tout le contenu de la disquette et de la revue Toolbox-Mag sont entièrement sous Copyright de Toolbox-Mag, et sont déposés à l'Agence pour la Protection des Programmes. Tous droits de traduction et de reproduction intégralement réservés pour tous pays. Aucune reproduction, même partielle, et sous quelque forme que ce soit, de la disquette comme de la revue Toolbox-Mag, ou d'un des programmes qu'elles contiennent, ne pourra avoir lieu sans accord préalable et écrit de Toolbox-Mag. Copier Toolbox-Mag, c'est illégal, et cela vise à torpiller le seul magazine français sur l'Apple // GS: nous serons alors sans pitié.


# Editorial (🍎)

## *Pour le plaisir, sans sectarisme, avec confiance*

Dès le numéro 2, nous réagissons aux souhaits de nos lecteurs: dans le flot des félicitations, nous avons cependant noté deux critiques, que nous reprenons à notre compte. Toolbox-Mag N°1 était un peu trop 'sérieux' dans son contenu, et ignorait le Pascal. Vous trouverez donc dans ce numéro une partie ludique plus conséquente, et plus de Pascal. Oui, c'est *pour le plaisir* que nous avons des GS.

L'entrée de quelques nouveaux membres à notre Conseil de Rédaction, que commente J.Y. Bourdin, témoigne d'une autre orientation de Toolbox-Mag: le *refus du sectarisme*. Tous les modes de programmation, même les plus bizarres, tous les langages, même les plus exotiques, ont leur place dans Toolbox-Mag, du moment qu'ils montrent ce qu'on peut faire avec un Apple II GS.
"Free Tools", "No Tools"? Bootez donc la disquette, et vous verrez ce que donne Toolbox-Mag en couleurs et en musique! Mais dans ce même numéro, un des auteurs de cette animation, membre éminent du "FTA" (Free Tools Association) publie... un Tool, un outil GS, eh oui ! Bravo Olivier Goguel (et Huibert Aalbers, l'auteur de SoundSmith) pour ce SoundTrack Tool.

Sans sectarisme, cela signifie aussi que nous demandons à tous nos lecteurs d'accepter la diversité des niveaux des articles et programmes de Toolbox-Mag. Il y en a pour chacun dans Toolbox-Mag, du débutant à l'auteur de Tools GS. Vous avez sans doute déjà repéré les 🍎 auprès des titres de nos articles et dans le sommaire: elles indiquent le niveau de difficulté de chaque article ou passage.
Débutants (🍎), il est normal qu'une partie de Toolbox-Mag vous soit incompréhensible pour le moment: mais on n'est pas éternellement débutant, surtout quand on a Toolbox-Mag pour apprendre (🍎🍎)... Si vous voulez bien jeter un coup d'œil sur les listings en Pascal de ce numéro, vous vous apercevrez que c'est plus facile que vous ne pensiez. Programmeurs et bidouilleurs (🍎🍎🍎), acceptez que votre langage ne soit pas le seul, et qu'il y ait aussi des parties élémentaires.

*"Il faut être gonflé pour sortir un magazine Apple II GS en ce moment"*, avez-vous dit: oui, nous sommes confiants. Bien entendu, nous entendrons encore toutes les rumeurs possibles sur ce serpent de mer de l'abandon de l'Apple II par Apple. Cette rengaine éculée ressort sans vergogne depuis des années et des années: nous sommes blindés. Pour rencontrer tant d'obstination dans le mensonge, et susciter tant de vœux de mort, c'est que nous devons sacrément déranger: *eh bien, ça va continuer, Messieurs!*
Ca va continuer, parce que la force principale de l'Apple II n'a jamais résidé chez Apple, mais dans ce qu'on ne rencontre que sur cette machine: *la communauté des utilisateurs*. Ca va continuer, parce que les lecteurs de notre magazine, qui savent déjà qu'ils en sont aussi les auteurs, seront également ses meilleurs agents publicitaires. Toolbox-Mag est bien démarré: il reste maintenant à chacun de nos lecteurs à abonner un de ses amis, et le pari sera gagné.

**Bien sûr que nous avons confiance: *en vous !***

Eric Weyland

# C Facile ! (••)

## Deuxième partie

François Uhrich

## Les Commandes Externes du Shell APW/Orca

La puissance de l'environnement APW est pour une bonne part dûe au nombre toujours croissant des outils de développement disponibles sous forme de commandes externes (autrement appelées utilitaires).
Ces commandes peuvent être des programmes de type EXE ou des textes de commandes Shell de type EXEC (langage interprété ressemblant au Shell script d'UNIX). Ils sont rassemblés dans le répertoire 6/ (UTILITIES par défaut).
Pour qu'APW puisse lancer ces programmes à partir d'un répertoire quelconque, ils doivent être décrits dans le fichier 4/SYSCMND où chaque ligne correspond à une commande interne ou externe d'APW.
Commande interne contenue dans le Shell APW.SYS16 (chaque commande interne possède un numéro unique):
*CAT C 4 catalog*
Commande externe non résidente:
*INIT U disk initializer*
Commande externe résidente (signe *):
*GET *U Saisie Clavier par François Uhrich août 1990*
Définition d'un langage (avec ici un compilateur résident):
*CC *L 8 ORCA/C Compiler*
(voir C Facile n°1 au sujet du numéro du langage).

Plusieurs sociétés ont déjà édité des ensembles d'utilitaires pour APW. Tout d'abord et bien évidemment *ByteWorks* avec son *Utility Package #1* (CAL, CHECK, CMP, LOWER, SORT, STRIPC, STRIPW, TCMP, TEE, UNIQ, UPPER, WC).
Il y a d'autre part les excellents produits de *360 MicroSystems* avec *File Utilities I* (ERASE, FIND, FLIST, HEAD, MORE, PERM, PRINT, REBOOT, SETDATE, SPLIT, STRIP, SUM, TAIL, TOUCH), *File Utilities II* (ARCV, GREP, MAKE et TRAPPER).

### Bravo Byte Works !

Dans Toolbox Mag n°1, j'avais mentionné comme un des atouts en faveur de Orca/C le suivi de Byte Works pour ces produits: tous les acheteurs enregistrés avaient en effet reçu la mise à jour 1.1 B11 en juin dernier.
Eh bien trois mois plus tard, c'est à dire en septembre, la **version définitive 1.1 de Orca/C** a été envoyée, toujours gratuitement, à ces mêmes heureux acheteurs !
A noter que dans cette version, les includes de la Toolbox du système 5.0 sont présents. La petite manipulation présentée dans mon précédent article n'est donc plus utile.

Bravo à Mike Westerfield et son équipe !

Certaines de ces commandes sont en provenance directe de l'environement UNIX, par exemple ARCV pour l'archivage, GREP pour la recherche (et en option le remplacement), dans des fichiers sources, d'une chaîne de caractères pouvant être donnée sous la forme d'une expression régulière complexe ou non.
La commande la plus indispensable à mon sens est MAKE. A partir d'un fichier texte appelé MAKEFILE par défaut, décrivant les interdépendances entre les fichiers sources, objets et exécutables, cette commande permet en effet de lancer automatiquement les compilations, les éditions de liens suivant les dates de dernière modification de ces fichiers. Voici un exemple de fichier MAKEFILE:

*MonProg.root: MonProg.cc MonProg.h*
*compile MonProg.cc keep=$*
*MonProg: MonProg.root 2/MaLib*
*link MonProg 2/MaLib keep=$*
*copy -c MonProg /SYSTEM.DISK/ MonProg*

Ainsi si MonProg.cc a été modifié, sa date de dernière modification est donc postérieure à celle de l'objet correspondant MonProg.root. Lorsque l'utilisateur lancera MAKE, ce dernier lira par défaut le MAKEFILE du répertoire courant et MAKE exécutera toutes les lignes de commandes liées à la dépendance entre MonProg.cc et MonProg.root, c'est à dire dans notre exemple: compile MonProg.cc keep=$.
Si une erreur survient lors de l'exécution d'une de ces commandes, MAKE stoppe son exécution et rend la main à l'utilisateur. Dans le cas contraire, maintenant que MonProg.root a été modifié et que donc ce fichier a une date postérieure à l'exécutable MonProg: c'est parti pour le link !
A noter que le MAKEFILE n'est pas parcouru séquentiellement mais hiérarchiquement: les lignes de dépendance peuvent donc être dans un ordre quelconque.
Cette commande n'est pas seulement utile (pour ne pas dire indispensable !) aux éternels cycles compilation-link mais, du fait de ce test de date, des backups automatiques peuvent être facilement réalisés.

## Ecrire en C des Commandes Externes APW

Une commande spécifique au Shell APW/Orca est lancée par l'utilisateur sous cette forme:
*# MonUtil [arg0 [ arg1... [argN]... ]]*
Par exemple:
*#TYPE MonTexte >.printer*
pour sortir sur imprimante le fichier MonTexte.

Le programme peut donc éventuellement nécessiter des arguments. La particularité de ces applications est qu'elles doivent être capables de relire les arguments de la ligne de commande du Shell.
Rien n'est plus simple en C: car UNIX est passé par là.
Tous les programmes C ont une fonction principale **main()** équivalente au bloc principal Begin..End en Pascal. Pour un programme classique (c'est à dire de type S16), main() n'a pas de paramètre. Pour un programme du Shell (de type EXE), main() peut avoir deux paramètres:
- *argc* : un entier indiquant les nombres de paramètres sur la

ligne de commande, incluant le nom du programme (donc argc est toujours supérieur à 0) et excluant toutes les redirections d'entrée-sortie (comme >.printer pour la sortie sur imprimante).
- *argv* : un pointeur de pointeurs de caractères, ou autrement dit un tableau de chaînes de caractères, qui correspondent à chacun des arguments de la ligne de commande.
Voici un exemple de lecture de la ligne de commande du Shell en C:

```
main(argc,argv)
int  argc;    /* nombre d'arguments */
char  *argv[]; /* tableau des arguments */
{
  int i;
  printf("Nombre d'argument(s) : %d\n",argc);
        for  (i=0;i<argc;i++)  printf("argument  %d  :
%s\n",i,argv[i]);
  exit(0);
}
```

Le programme affiche tout d'abord le nombre d'arguments argc puis un à un tous les arguments.
Remarquez le exit(0) à la fin: il est là pour indiquer au shell que l'exécution s'est bien déroulée. Si une erreur survient et que le programme doit être abandonné, il faut quitter avec exit(-1) (tout entier différent de 0). Cela est très important car si vous utilisez votre programme dans un fichier EXEC et que vous ne quittez pas par un exit, le Shell recevra une valeur quelconque (donc probablement différente de 0) indiquant une erreur et stoppera l'exécution de l'EXEC. La valeur retournée par une commande Shell ou un EXEC peut être retrouvée dans la variable shell STATUS. A noter que dans la fonction main(), exit(n) est équivalent à return(n).
Remarquons que ce petit exemple simpliste peut être aisément écrit en langage interprété EXEC:

```
echo {#}
echo {0}
for i
echo {i}
end
```

La variable {#} contient le nombre d'arguments de la commande, celle-ci exclue: # est donc égal à argc-1.
{0} contient la chaîne de la commande, c'est à dire argv[0].
Plus généralement {i} contient argv[i].

## Enfin le source du GET !

Voilà une démonstration nettement plus intéressante de l'utilisation de argc et argv. Voici en effet le source de la commande GET donnée dans Toolbox-Mag n°1.
Sur la disquette vous retrouverez ce listing, avec en plus tout ce qu'il faut pour pouvoir également compiler le même source avec APWC, en profitant de l'utilisation des commandes du pré-processeur (ligne commençant par #) avec la macro __ORCAC__ qui vaut 1 si le compilateur est bien Orca/C et 0 sinon (donc avec APWC...). Attention toutefois au langage du source qui doit être approprié au compilateur utilisé (voir la commande CHANGE décrite dans C Facile 1).

```
#include  <textitool.h>
#include  <shell.h>
#include  <string.h>

main(argc,argv)
int  argc;
char  *argv[];
{
  char  c[2];
  Set_VarPB var;

  if ((argc!=2)&&(argc!=3)) {
    WriteLine("\pUsage : Get varShell [Prompt]");
    exit(-1);
  }
  if (argc==3) WriteCString(argv[2]);
  c[0]=1;
  c[1]=(char)ReadChar(0);
  var.var_name = c2pstr(argv[1]);
  var.value = c;
  SET_VAR(&var);
  exit(0);
}
```

### Là où il y a de la chaîne...

Pour les débutants, une des difficultés de la programmation en C de la Toolbox provient de l'utilisation de différents types de chaînes de caractères:
- en C standard, c'est la chaîne C qui est utilisée. Cette chaîne se termine par un caractère nul.
- en Pascal et dans la Toolbox, c'est la chaîne Pascal qui prime. Cette chaîne commence par sa longueur sur un octet (ce dernier n'étant pas compté dans la longueur) puis suivent les caractères. Comme la longueur est sur un octet, la chaîne Pascal est limitée à 255 caractères.
Pour simplifier la vie des programmeurs C, APWC et Orca/C offrent un moyen simple pour ne pas avoir à compter les caractères d'une chaîne Pascal: il suffit de commencer la chaîne par \p comme dans "\pUsage : Get varShell [Prompt]". Le compilateur remplacera ce symbole par l'octet de la longueur. Ouf !
- depuis l'arrivée de GS/OS, un nouveau type de chaîne vient encore compliquer les choses. Les noms de fichiers en ProDOS8 et ProDOS 16 sont des chaînes Pascal donc avec un octet pour la longueur. GS/OS émule ProDOS16 avec ses appels de type Classe 0 et donc à l'aide de chaînes Pascal.
Mais les appels spécifiques à GS/OS de Classe 1 nécessitent des chaînes dites de Classe 1: la chaîne débute comme auparavant par la longueur mais celle-ci est stockée sur un mot (2 octets), les caractères suivent. Cette longueur sur 2 octets permet l'utilisation de chemins d'accès très longs...

## printf(format,arg1,arg2, ... )

printf convertit, met en forme et imprime ses arguments sur la sortie standard (par défaut l'écran) suivant le format donné. Cette chaîne format est composée de deux types d'objets: d'une part les caractères ordinaires recopiés sans modification vers la sortie standard, et d'autre part les spécifications de conversion et d'impression de chacun des arguments arg1,arg2...

Chaque spécification débute par le caractère % et se termine par un caractère de conversion: d ou i décimal, u décimal non signé (unsigned), o octal, x ou X hexadécimal (si x alors abcdef, si X ABCDEF), c caractère, s chaîne, f réel (float)...etc. Pour écrire un simple % par printf, il faut écrire %%...

Entre ces deux caractères, on pourra trouver entre autres un nombre précisant la largeur minimale du champ de l'argument. Par exemple (\n signifie un retour à la ligne):

```
printf("Hello World\n");
printf("Nombre: %4i  Chaine: %s\n",i,str);
printf("x=%10f sin(x)=%10f cos(x)=%10f\n",x,sin(x),cos(x));
```

Cette instruction, on le voit, est extrêmement puissante. Mais elle est très coûteuse en code, surtout avec APWC qui va carrément placer dans le programme un interpréteur du printf! En Orca/C, le printf est moins gourmand, mais je recommande tout de même l'utilisation dans la mesure du possible du Text Tool (avec notamment WriteCString et WriteLine dans GET). Cela bien sûr uniquement dans le cas de programmes spécifiques au GS, qui ne seront pas portés en C sur d'autres machines: entrent dans cette catégorie les NDA, CDA, CDev et Inits, dont la taille doit être la plus réduite possible.

texttool.h contient la déclaration de WriteLine, WriteCString et ReadChar; shell.h pour SET_VAR et la structure Set_VarPB; enfin string.h pour c2pstr.

Tout d'abord, GET vérifie qu'il y a bien 2 ou 3 paramètres sur la ligne de commande APW; sinon il affiche un message d'erreur et quitte avec exit (-1).

Puis s'il y a 3 paramètres, il affiche le texte du 3ème paramètre argv[2].

ReadChar(0) attend et retourne la valeur de la touche frappée par l'utilisateur (0 signifie sans écho à l'écran).

Le nom et la valeur d'une variable du Shell sont des chaînes de caractères de type Pascal. La valeur de la variable Shell donnée en paramètre au GET sera une chaîne de longueur 1 puisqu'il s'agit d'une touche du clavier d'où le c[0]=1.

Le nom de la variable est contenu dans argv[1] sous la forme d'une chaîne C; donc il faut la convertir en chaîne Pascal... Heureusement la librairie C nous offre les services de fonctions de conversion: *c2pstr* et *p2cstr*. Attention toutefois car le résultat de ces deux fonctions est stocké dans un buffer interne, lequel est donc modifié à chaque appel: il est en conséquence recommandé de recopier le résultat de ces fonctions dans une chaîne à soi...

Dans notre cas, c'est la fonction SET_VAR qui s'en charge: cette fonction permet l'affectation d'une valeur à une variable Shell. Elle nécessite comme paramètre l'adresse (d'où le &) d'une structure Set_VarPB composée de 2 pointeurs de caractères (c'est à dire d'adresses de chaînes):

```
typedef struct
 {
  char *var_name;          /* variable name */
```

## for (expression1;expression2;expression3) instruction;

Cette instruction est équivalente à:

```
expression1;
while (expression2)
{
instruction;
expression3;
}
```

expression1 précise l'initialisation de la boucle; expression2 indique le test effectué avant chaque itération; enfin expression3 est évaluée à chaque itération et contient habituellement l'incrémentation ou la décrémentation d'une variable utilisée dans le test expression2.

Par exemple, voici quatre façons de faire la même initialisation d'un tableau de 10 entiers:

```
int i, tab[10];
for (i=0;i<10;i=i+1) tab[i] = 0;
for (i=0;i<10;i++) tab[i] = 0;
for (i=0;i<10;tab[i++]=0);
i=0;
while (i<10)
{
tab[i]=0;
i++;
}
```

```
char *value;            /* variable value */
 }
Get_VarPB, Set_VarPB;
```

SET_VAR n'est pas un appel Toolbox et encore moins un appel de la librairie. Le Shell APW contient en effet près d'une trentaine de primitives dont SET_VAR fait partie (voir l'include Shell.h et la documentation APW). Ces fonctions ne peuvent bien sûr être utilisées que si le Shell est présent, et donc uniquement à l'intérieur de programmes de type EXE.

## Commande Purge

Pour finir, vous trouverez sur la disquette une autre commande externe destinée au Shell APW: **Purge**. Elle effectue la purge la plus complète possible de la RAM du GS. Cela peut être utile lorsqu'on utilise plusieurs langages sous APW, car la plupart des compilateurs sont résidents en mémoire après leur premier chargement, sans compter les includes C ou equates Assembleur qui sont également conservés; au bout d'un certain temps, la mémoire peut donc être pas mal encombrée.
De plus avec la version actuelle plutôt défaillante de la commande Duplicate (jointe avec APW Tools and Interfaces v1.1) qui permet la copie de fichier avec ressources, la purge est recommandée avant toute utilisation.
L'option (-p) permet d'afficher des statistiques quant à l'occupation mémoire avant et après la purge. Il s'agit du même affichage que dans mon NDA TransProg informant l'utilisateur de la mémoire totale, libre, réellement libre (si la purge est déclenchée) et enfin le bloc maximal de mémoire libre.
Cet affichage optionnel occupe la quasi totalité du source car la purge elle-même est réalisée par une unique instruction:
*NewHandle (TotalMem(),0x1000,0,0L);*
Il s'agit d'une demande d'allocation mémoire au Memory Manager. La particularité de cette demande est de générer obligatoirement une erreur, puisque la quantité de mémoire demandée est la mémoire totale présente, qui bien sûr ne peut pas être allouée dans sa totalité puisqu'une partie est déjà occupée! Mais l'intérêt est que le Memory Manager va quand même voir s'il n'est pas possible de satisfaire cette demande: d'une part les blocs mémoires purgeables vont être libérés, d'autre part les blocs mémoires non purgeables vont être déplacés pour obtenir le bloc mémoire libre le plus grand possible.

J'espère que cet aperçu des facilités d'écriture d'utilitaires APW en C vous donnera envie d'en écrire vous-même. Dans ce cas n'hésitez pas à nous envoyer les plus intéressants !

Byte Works, 4700 Irving Blvd. NW, Suite 207
Albuquerque, NM 87114, USA

360 Microsystems, 12272 Fox Hound Lane
Orlando, FL 32826, USA

## Les bouche-trous de la Rédaction

Aussi experte que soit notre maquettiste, il reste toujours, dans une revue essentiellement faite par ses lecteurs comme Toolbox-Mag, des "trous" dans les fins d'articles. Or, dans Toolbox-Mag, il n'y a pas de publicités pour boucher ces trous.

On peut s'en tirer en "sacrifiant" un article, un listing par exemple, qu'on tronçonne en plusieurs morceaux (suite du listing page 12, suite du listing page 27, etc). Comme lecteur, je déteste cette formule.

Nous essayons donc dans ce numéro une autre solution: les "bouche-trous". Par définition, ces bouche-trous n'auront guère de place pour les nuances et les arguments: il faudra donc les prendre avec une pincée de sel. Dîtes-nous cependant si cette formule vous plait.

Dans ce numéro, vous trouverez aussi en bouche-trou des "brèves", sous le titre "Entendu à l'Expo". Peu importe qui les a dites: nous les avons entendues à l'Apple Expo, et elles sont aussi drôles que certaines "brèves de comptoir". Nous les avons notées, et nous vous les livrons telles quelles.

JYB

## Programmer le GS.

Cette question m'a été posée assez souvent à l'Expo: que recommandez-vous pour programmer le GS? Ma réponse est un peu biaisée: le langage dans lequel vous programmez importe peu, et tout est bon, du moment que l'interface avec la Toolbox du GS est propre. Cela exclut la plupart des Basic, mais inclut le Pascal (TML, Orca), le C (APW, Orca), l'Assembleur (APW/Orca, Merlin, Lisa 8-16).

Si vous n'avez pas de préférence au départ, je recommande, au choix, Orca-Pascal ou Orca-C, de ByteWorks, à acquérir avec le cours de Pascal ou de C (de vrais cours, très bien faits, avec interrogations écrites et tout) que ByteWorks diffuse en même temps. Pour l'Assembleur, Merlin ou APW avec le livre de Wagner. Joignez-y Genesys ou Design Master pour ne pas perdre de temps sur l'interface utilisateur.

Je recommande essentiellement la lecture d'un nouveau magazine GS: Toolbox-Mag! Sans rire: d'une part, nous veillons à ce que nos auteurs s'adressent aussi aux débutants. Surtout, la meilleure manière d'apprendre, c'est d'éplucher les programmes des autres.

Il y a et il y aura des listings dans les colonnes de Toolbox-Mag: loin d'être, comme le suggère un de nos lecteurs, "une façon artificielle de gonfler le nombre de pages", les listings sont notre manière de répondre à vos demandes d'initiation et de formation. Car programmer, c'est écrire des listings. Ceci dit, une partie de ces listings sera sur la disquette: imprimez-les et épluchez-les aussi.

Sachez enfin qu'il y a deux préalables à la programmation du GS: 1/ Il vous faut une configuration qui tienne la route (2 Mégas de mémoire et disque dur). 2/ Il faut, pour citer l'un d'entre vous, "s'appuyer la Toolbox".

Programmer le GS, quel que soit le langage, c'est s'interfacer avec sa Boîte à Outils. Les trois tomes du Toolbox Reference sont indispensables, il n'y a pas de raccourci. Le début est ingrat. Mais très vite, vous verrez que le GS fait tout tout seul, et c'est bien confortable.

Si vous préférez faire de la programmation "No Tools", alors il va falloir "vous appuyer" au moins le HardWare Reference et le Firmware Reference. Et il faudra tout faire vous-même.

JYB.

# SoundTrack Tool

**Un outil pour jouer les musiques SoundSmith (♯♯♯)**

Olivier Goguel

*SoundSmith, de Huibert Aalbers, est un des meilleurs logiciels musicaux pour l'Apple II GS. Le programme séquenceur (utilisé ici dans sa version 0.95) et de nombreuses disquettes de musiques et d'instruments sont disponibles en freeware en France, en shareware ailleurs, donc auprès de tous les bons grille-pains. Le SoundTrack Tool, ou Outil 219, ou ST, vous permettra de jouer les musiques SoundSmith à l'intérieur de vos programmes.*

*La documentation ci-dessous se limite à décrire l'interface utilisateur (appels et erreurs) de l'outil. Vous trouverez des compléments dans le source de l'exemple en assembleur. Relisez aussi l'article de Stephan Hadinger dans Toolbox-Mag N°1 : je me suis dispensé de le répéter. Le source de l'outil lui-même sera publié dans un prochain Toolbox-Mag, ce qui permettra de le mettre à jour pour la dernière version de SoundSmith. En attendant, chacun pourra s'en servir "en aveugle", comme d'un vulgaire outil Apple.*

*J'ai choisi que l'outil s'occupe lui-même du chargement des musiques et de libérer la mémoire ensuite. Quitte à faire un outil, autant qu'il fasse le travail. J'ai rajouté une Unit et les interfaces pour TML-Pascal, voir ci-dessous. Mais il reste des choses à faire: des macros pour Merlin et APW/Orca, l'interface avec le C et avec Orca-Pascal, sans compter les Help de Merlin, les compléments pour le fichier de Nifty List et celui d'Orca-Disasm, voire le Programmer's Companion. Il faut bien laisser un peu de travail aux lecteurs de Toolbox-Mag, non? La disquette Toolbox-Mag attend vos fichiers.*

*Vous trouverez sur la disquette un fichier texte GS appelé "USER.TOOL". J.Y Bourdin y explique pourquoi nous avons choisi de ne pas en faire un "User Tool". Quitte à faire un outil, autant en faire un vrai...*

***Bibliographie:***


- ♣ Music Juke-Box, par Stéphan Hadinger, Toolbox-Mag N° 1
- ♦ Documentation du logiciel SoundSmith
- ♥ Apple II GS Toolbox Reference, Vol 2 et 3
- ♠ TechNote Apple, File Type Note, FTN.D5.0007


---

## *Description de l'outil 219 ($DB): SoundTrack Player ou ST.*

***STStartup*** ***($02DB)***

Entrée : MemId: Word
Sortie : -
Erreur : $DB01, ST déjà initialisé.

Cet appel initialise ST, et en particulier met en route une interruption sonore sur l'oscillateur no 30 ($1E) permettant par la suite de réguler le tempo. Il est conseillé de faire cet appel au moment voulu afin de ne pas être dérangé par cette interruption.
A noter que ST n'est pas compatible avec les autres outils gérant le son, puisqu'il s'approprie la routine d'interruption ainsi que tous les oscillateurs...

***STShutDown.*** ***($03DB)***

Entrée : -
Sortie : -
Erreur : -

Cet appel arrête la musique en cours, libère toute la mémoire allouée pour les musiques, remet le vecteur d'interruption sonore à sa valeur initiale, et arrête l'interruption donnant le tempo.

***STVersion.*** ***($04DB)***

Entrée : -
Sortie : N° Version: Word
Erreur : -

Retourne le numéro de version de l'outil, actuellement $0100 = 1.0.

***STStatus.*** ***($06DB)***

Entrée : -
Sortie : Statut: Boolean
Erreur : -

Renvoie True ($0001) si l'outil a déjà été initialisé ou ($0000) sinon.

***STLoadOneMusic.*** ***($09DB)***

Entrée : FileName: Pointer
Sortie : -
Erreur : $DB02, ST non initialisé
$DB04, Format musique incorrect
$DB09, Format WvBank incorrect

Charge la musique en mémoire ainsi que la wavebank correspondante. La wavebank est définie par le même nom de fichier que la musique, auquel on ajoute le suffixe .W
La wavebank est créée à partir de Soundsmith avec l'option Save SoundFile, *en utilisant le format GS Soundsmith* (et surtout pas Music Studio !).
Une fois que le fichier wavebank est chargé, la musique est initialisée, la wavebank est copiée dans la ram son, puis la mémoire allouée pour la wavebank est libérée.
Cet appel est prévu surtout pour le cas où on n'a qu'une seule musique à jouer. Sinon, il faudra utiliser l'option Batch.
A noter: les fichiers Wavebank issus de SoundSmith ne sont pas pourvus d'une signature fiable. L'outil ne peut donc vérifier qu'il s'agit bien d'une Wavebank. L'erreur "Format Wavebank incorrect" ne sera donc jamais renvoyée pour le moment: mais cela viendra peut-être un jour.
En attendant, vérifiez bien que votre Wavebank est au format GS Soundsmith, et pas Music Studio!

## Question de droits

Comme tous les programmes Toolbox-Mag, le Tool 219 est librement utilisable, pour leur utilisation personnelle, par les abonnés à Toolbox-Mag, et par eux seulement.
Mais vous pouvez avoir envie d'incorporer cet outil dans des programmes que vous diffuserez à d'autres: seulement, là, il y a le Copyright et les droits. Diffuser un programme de Toolbox-Mag auprès d'autres que vous, c'est faire de la copie illégale.
Donc, si votre programme est diffusé en freeware, ne mettez pas le Tool 219 sur votre disquette: à la place, donnez l'adresse de Toolbox-Mag, pour que l'utilisateur puisse l'acquérir.
Si votre programme est vendu, sous quelque forme que ce soit y compris le shareware, contactez Toolbox-Mag: il y aura des droits à payer avant de mettre le Tool 219 sur votre disquette. Ne faîtes rien sans notre accord préalable et écrit.

Toolbox-Mag

## Pour le débutant (🍎)

Si vous voulez simplement pouvoir voir et écouter les programmes et musiques de démonstration de ce disque, voici ce qu'il faut faire:
1/ Copiez le fichier intitulé TOOL219 depuis le sous-catalogue /SOUNDSMITH.TOOL de la disquette /TOOLBOX.MAG.02A, sur votre disque système, dans le sous-catalogue /SYSTEM/TOOLS.
2/ Passez sous le Finder, introduisez la disquette /TOOLBOX.MAG.02A, double-cliquez sur l'icône intitulée ST.EXEMPLE.ASM.

***STPlayMusic.*** ***($0ADB)***

Entrée : LoopMode: Boolean
Sortie : -
Erreur : $DB02, ST non initialisé
$DB03, Pas de musique initialisée

Rejoue la dernière musique sélectionnée. Il peut s'agir de la dernière musique chargée avec l'outil LoadOneMusic ou SelectBatchMusique.
Le paramètre booléen LoopMode permet d'indiquer à l'outil s'il doit rejouer la musique dès qu'elle est terminée ($0001 = True), ou au contraire s'il doit s'arrêter ($0000 = False)

Ex: charger et jouer une musique qui boucle

En Assembleur :

```
pea  ^FileName
     pea  FileName
     ldx  #$09DB
     jsl  $E10000     ; _STLoadOneMusic
     _Err
     pea  $0001
     ldx  #$0ADB
     jsl  $E10000     ; _STPlayMusic
     _Err
     ...
FileName str  '9/MusicFile'
```

En Pascal :

```
STLoadOneMusic('9/MusicFile');
STPlayMusic(True);
```

***STStopMusic.*** ***($0BDB)***

Entrée : -
Sortie : -
Erreur : -

Arrête la musique en cours.

***STGetEndOfMusic.*** ***($0CDB)***

Entrée : -
Sortie : FinishedFlag: Boolean

## Le Tool 69 aussi

Vous trouverez sur la disquette, dans le sous-catalogue /MJUKE.TML.PAS, une Unit d'interface de l'outil 69 de Stephan Hadinger avec TML Pascal. C'est Mr Destelle, un expert en TML Pascal et en ressources, qui nous l'a envoyée.

Nous vous recommandons de lire le fichier MJUKE: il contient des remarques importantes de Mr Destelle sur l'utilisation du Tool 69 sous TML Pascal.

Erreur : -

Renvoie Vrai ($0001) dès que la musique arrive à son terme (même si elle boucle).

***STAddToBatch. ($0DDB)***

Entrée : FileName: Pointer
BatchNb: Integer
Sortie : -
Erreur : $DB02, ST non initialisé.
$DB05, Il existe déjà une musique au numéro indiqué.
$DB06, Mauvais numéro de musique.
$DB09, Format WaveBank incorrect

Charge une musique en mémoire en lui donnant un numéro permettant de la jouer au moment voulu. Le Numéro de la musique doit être compris entre 1 et 25.
Cet appel charge en mémoire la musique correspondante ainsi que sa wave (.W), mais, contrairement à l'appel LoadOneTool, n'initialise pas la musique. Celle-ci devra être sélectionnée par l'appel STSelectBatch pour pouvoir être jouée.
Même remarque que ci-dessus en ce qui concerne l'erreur "Wavebank incorrecte".

***STSelectBatch. ($0EDB)***

Entrée : BatchNb: Integer
Sortie : -
Erreur : $DB06, mauvais numéro de musique
$DB07, aucune musique ne correspond au numéro indiqué.

Sélectionne la prochaine musique à jouer. Cet appel arrête la musique en cours et libère éventuellement la mémoire allouée pour la musique chargée par l'appel LoadOneMusic. (Il est évidemment déconseillé d'utiliser à la fois l'appel STLoadOneMusic et STAddToBatch...)
Ex : Charge et joue plusieurs musiques.

En Assembleur:

### Unit "ST" Pour TML Pascal

```
UNIT ST;
INTERFACE

Const MaxBatch = 25;
Type TPtrTListe = ^TListe;
TListe = Array [1..MaxInt] of Integer;
TPtrTListeVu=^TListeVu;
TListeVu = Array [1..32] of Integer;

PROCEDURE   LoadOneTool          (Num,Version : Integer);              Tool 1,15;
PROCEDURE   STStartup            (MemID : integer);                    Tool 219,2;
PROCEDURE   STShutDown;                                                Tool 219,3;
FUNCTION    STVersion :          Integer;                              Tool 219,4;
FUNCTION    STStatus :           Integer;                              Tool 219,6;
PROCEDURE   STLoadOneMusic       (Filename : string);                  Tool 219,9;
PROCEDURE   STPlayMusic          (LoopMode: Boolean);                  Tool 219,10;
PROCEDURE   STStopMusic;                                               Tool 219,11;
FUNCTION    STGetEOfMusic:       Boolean;                              Tool 219,12;
PROCEDURE   STAddToBatch         (Filename : string; BatchNb : Integer); Tool 219,13;
PROCEDURE   STSelectBatch        (BtachNb : Integer);                  Tool 219,14;
PROCEDURE   STKillBatch          (BtachNb : Integer);                  Tool 219,15;
FUNCTION    STGetPlayingMusic :  Integer;                              Tool 219,16;
PROCEDURE   STPlayBatch          (ListeMus : TPtrTListe);              Tool 219,17;
FUNCTION    STGetTrackVu :       TPtrTListeVu                          Tool 219,18;
PROCEDURE   STPauseMusic;                                              Tool 219,19;
PROCEDURE   STContinueMusic;                                           Tool 219,20;

IMPLEMENTATION
END.
```

```
pea ^FileName1
    pea  FileName1
    pea  £1            ; Musique No1
    ldx  #$0CDB
    jsl  $E10000       ; _STAddToBatch
    _Err
    pea  ^FileName2
    pea  FileName2
    pea  £2            ; Musique n°2
    ldx  #$0CDB
    jsl  $E10000       ; _STAddToBatch
    _Err
```

En Pascal :

```
STAddToBatch('9/Music1',1)
STAddToBatch('9/Music2',2);
STSelectBatch(1);
STPlayMusic(False);
While not STGetEndOfMusic;
STSelectBatch(2);
STPlayMusic(False);
```

***STKillBatch.*** ***($0FDB)***

## Un exemple en TML Pascal

*NB: cet exemple étant exécutable en mode texte, il faut taper des "{" sous TML pour avoir des "é" sur l'écran.*

```
Program ExempleTML;
Uses ST;
Var Liste : TPtrTListe;
ListeVu : TPtrTListeVu;
Car : char;
Begin
    New(Liste);
    Writeln('D{monstration du TOOL219');
    Writeln('Ecrit par O.GOGUEL.');
    Writeln;Writeln('(C)opyright FTA & ESP,
    Octobre 1990');
    Writeln;Writeln('(C)opyright Toolbox-Mag,
    Octobre 1990');Writeln;
    Writeln('Chargement des outils');
    LoadOneTool(219,0);
    Write('Ins{rez le disque syst}me si n{cessaire:');
    Writeln; Readln(car);
    STStartup(57005);
    Writeln('Musique :'); Writeln;
    Writeln('Chargement de MODULAE
    par L.PICAMELOT');
    STAddToBatch ('9/Music/Modulae',1);
    Writeln('Chargement de MODULAE INTRO
    par S.MOUSSERON');
    STAddToBatch ('9/Music/Intro',2);
    Writeln('Chargement de TOOLBOX SONG
    par L.PICAMELOT');
    STAddToBatch ('9/Music/Toolbox',3);
    Liste^[1]:=1;Liste^[2]:=2;Liste^[3]:=3; Liste^[4]:=-1;
    STPlayBatch(Liste);
    Writeln;Writeln;
    Writeln('Des musiques pour le plaisir...');
    Readln(Car);
    STShutDown;
End.
```

```
pea  £1
     ldx  £$0DDB       ; _STSelectBatch
     jsl  $E10000
     _Err
     pea  £0           ; False
     ldx  £$0ADB       ; _STPlayMusic
     jsl  $E10000
     _Err              ; Joue musique n°1
§lp  pea  £0
     ldx  £$0CDB       ; _STAddToBatch
     jsl  $E10000
     pla
     beq  §lp          ; boucle tant que la
                       ; musique n'est pas finie...
     pea  #2
     ldx  £$0DDB       ; _STSelectBatch
     jsl  $E10000
     _Err
     pea  £0           ; False
     Ldx  £$0ADB       ; _STPlayMusic
     jsl  $E10000
_Err                   ; Joue musique n°2

FileName1 str '9/Music1'
Filename2 str '9/Music2'
```

Entrée : Batch Nb: Integer
Sortie : -
Erreur : $DB02, ST non initialisé
$DB06, mauvais numéro (de 1 à 25)
$DB07, pas de musique à ce numéro

Libère la mémoire allouée à une musique donnée ainsi que sa place dans la liste des musiques. Arrête la musique en cours si elle correspond à la musique donnée.
Si on donne 0 comme numéro de musique, toutes les musiques chargées avec l'option batch sont enlevées de la liste, et les mémoires correspondantes libérées.

***STGetPlayingMusic.*** ***($10DB)***

Entrée : -
Sortie : Numéro de musique: Integer
Erreur : -

Renvoie -1 ($FFFF) si aucune musique n'est en train d'être jouée, 0 si c'est une musique chargée avec l'option LoadOneMusic, et sinon son numéro dans la liste batch.

***STPlayBatch.*** ***($11DB)***

Entrée : Liste: Pointer

Sortie : -
Erreur : $DB02, ST non initialisé
$DB08, Mauvaise séquence...

Permet de gérer automatiquement l'enchaînement de musiques. Pour cela il suffit de fournir à l'outil l'ordre dans lequel on veut jouer les musiques chargées préalablement avec l'option Batch.
Un 0 dans la liste indique sa fin, tandis qu'un -1 ($FFFF) fait boucler l'enchanement.

Ex : Jouer les musiques 1 puis 2, puis 2 encore puis 3, puis recommencer...

En Assembleur :

```
pea   ^BatchList
      pea   BatchList
      ldx   £$11DB
      jsl   $E10000
      _Err

BatchList da 1,2,2,3,$FFFF
```

En Pascal, on utilisera un pointeur de tableau d'entiers pour donner la liste en paramètres à l'outil. Pour cela, le type TPtrTListe est déjà défini dans l'Unit ST.

```
Program Soundtrack;
Use ST;
Var Liste : TPtrTListe;
Begin

STStartup(Id);
New(Liste);
...
STAddToBatch('9/Music1',1);
STAddToBatch('9/Music2',2);
STAddToBatch('9/Music3',3);

Liste^[1]:=1;Liste^[2]:=2;Liste^[3]:=2;
Liste^[4]:=3;Liste^[5]:=-1;
STPlayBatch(Liste);
...
STShutDown;
End.
```

***STGetTrackVu.*** ***($12DB)***

Entrée : -
Sortie : Adresse TableVumètre: Pointeur
Erreur : -

Retourne l'adresse en RAM d'une table donnant un pseudo-volume sur 16 bits (valeur comprise entre $0000 et $00FF) de chacun des tracks et de chacun des instruments utilisés. Les $20 premières valeurs correspondent aux 14 tracks (2 valeurs inutilisées) et les $20 suivantes au volume de chacun des instruments.
La valeur retournée pour chacun des tracks ou instruments n'est pas utilisée dans l'outil, et peut donc être modifiée à tout moment sans aucune incidence sur la musique; elle est mise à jour dès qu'une nouvelle note est jouée. Cette table permet de faire afficher sur l'écran des vumètres correspondant soit aux tracks, soit aux instruments.
Voir le programme de démonstration en assembleur pour un exemple de vumètre en interruption.

***STPauseMusic.*** ***($13DB)***

Entrée : -
Sortie : -
Erreur : -

Permet d'interrompre l'exécution de la musique.

***STContinueMusic.*** ***($14DB)***

Entrée : -
Sortie : -
Erreur : -

Relance une musique à partir de l'endroit où elle avait été arrêtée avec l'outil STPauseMusic.
NB: l'outil ne renvoie pas d'erreur pour ces appels. C'est donc au programmeur de faire attention à ne pas continuer une musique inexistante.

---

## SoundSmith Tool : Un exemple en assembleur Merlin 16

*NB: Ce listing n'utilise pas de macros pour les appels outils, pour assembler plus vite. Mais le nom des macros est en commentaire.*
*On notera que les affichages se font avec une routine spécifique, et pas avec QuickDraw: No Tools...*

```
* Programme de Démonstration de l'Outil 219
* (C) FTA & ESP, Octobre 1990
* (C) Toolbox-Mag, Octobre 1990
* Assemblage sous MERLIN 16 avec LINKER.GS
* Tapez Pomme Ouverte 6 (le 6 du clavier, pas du pavé)
      lst     off
      rel
      dsk     ST.EXEMPLEASM.L
*------------------------------------------ Macros
_Err MAC                  ; Gestion des Erreurs
      BCC     NoErr
      DO      §0          ; Si un nom de message
      ldx     £^Here      ; est spécifié, on l'utilise
      ldy     £Here
      brl     PgmDeath
Here STR      §1
      ELSE
      brl     PgmDeath0   ; Sinon, message par défaut
      FIN
NoErr EOM
*------------------------------------------ Paramètres du VUMètre
VULgn =       70
VUCol =       60
VUNbTrack = 14
*------------------------------------------ Let's go...
Start_Pgm     clc
      xce
      rep     $30
      phk
```

```
      plb

      sep   $20

§lp   ldal  $E1C02E    ; Attend que le spot ne soit
      cmp   £$F8       ; plus sur l'écran pour éviter
      bcc   §lp        ; des barres de synchro

      lda   £$41       ; Passage en mode texte
      stal  $E1C029
      ldal  $E1C034    ; Sauvegarde la bordure
      sta   Save_C034  ;
      lda   £0         ;
      stal  $E1C034    ; Bordure Noire
      ldal  $E1C022    ; Sauvegarde la couleur du texte
      pha              ; dans la pile
      lda   £0         ;
      stal  $E1C022    ; Texte + Fond Noirs

      rep   $20

      ldx   £0
      txa
§lp   stal  $E12000,X  ; Efface
      inx              ; Ecran + SCB + Palettes
      inx
      bpl   §lp
                       ; Création Palettes
                       ; pour le VUMètre
      ldx   £0         ;
      lda   £$F00      ; dégradé du Rouge au
§lp   stal  $E19E02,x  ; Jaune
      pha
      lda   £$0FFF     ; Blanc
      stal  $E19E1E,x
      lda   £$0F00     ; Rouge
      stal  $E19E1C,x
      txa
      clc
      adc   £$20
      tax
      pla
      clc
      adc   £$010
      cpx   £$200
      bcc   §lp

      sep   $30        ; Positionne les
                       ; SCB pour le VUMètre
      ldx   £0
§lp   txa              ; Dégradé en 16 couleurs
      lsr              ; du rouge au jaune sur 32 lignes
      stal  $E19D00+VULgn+1,x
      inx
      cpx   £32
      bcc   §lp

      rep   $30

      ldx   £5*$A0+4   ; Affichage du Texte
      jsr   Print      ; Cf Print
      hex   0CFF
      asc   '      DEMONSTRATION TOOL219.',8D8D0CEE
      asc   '       SOUNDSMITH PLAYER.',8D0CFF
      hex   8D
      asc   '      ECRIT PAR O.GOGUEL.',8D
      asc   ' (C)OPYRIGHT FTA & ESP, OCT 1990.',8D
      asc   ' (C)OPYRIGHT TOOLBOX-MAG, OCT 1990.',00

      ldal  $E10028    ; Détourne le vecteur
      sta   Save28     ; SCBIrq pour l'envoyer
      dfb   $A9,$5C,<Inter ; sur notre routine de
      stal  $E10028    ; VuMètre...
      ldal  $E1002A    ;
      sta   Save2A     ; On aurait pu utiliser une
      dfb   $A9,>Inter,^Inter ; autre source d'interruptions...
      stal  $E1002A

      sep   $20        ; Attend de nouveau
§lp   ldal  $E1C02E    ; le passage du spot
      cmp   £$F8
      bcc   §lp
      lda   £$C1       ; Passage en mode graphique
      stal  $E1C029
      ldal  $E1C023
      and   £2!$FFFF
      stal  $E1C023
      pla
      stal  $E1C022    ; Récupère couleur du texte maintenant
      rep   $20        ; que nous sommes en page graphique

      pea   £219       ; Chargement de l'outil no 219
      pea   £0         ; No Version requis = 0 min.
      ldx   £$0F01     ; _LoadOneTool
      jsl   $E10000
      _Err  "Impossible de charger le Tool219 : $"

      pea   $DEAD      ; UserId (Why not ?)
      ldx   £$02DB
      jsl   $E10000    ; _STStartup
      _Err

* Chargement des musiques

      lda   £1         ; On va commencer par la musique no1
      sta   Playing_Music

§More_Load jsr Print_Loading

      lda   Playing_Music
      dec
      asl
      asl
      tay
      lda   Parm_Music,y
      jsr   Print_Rout ; Affiche le nom de la musique

§Reload lda Playing_Music
      dec
      asl
      asl
      tay
      pea   £^Parm_Music
      lda   Parm_Music+2,y
      pha
      lda   Playing_Music
      pha
      ldx   £$0DDB
      jsl   $E10000    ; _STAddToBatch
      cmp   £$45
      bne   No_MountErr

                       ; Si on a une erreur 45, c'est que
                       ; le chargement de l'outil219 nous
      ldx   £190*$A0+35 ; a fait changer de disquette...
      jsr   Print
      hex   0CEE
      asc   "INSEREZ LE DISQUE PROGRAMME...",00
      sep   $20
      stal  $E1C010
§lp   ldal  $E1C000    ; On redemande le disque programme
      bpl   §lp
      rep   $20
      ldx   £190*$A0+35
      jsr   Print
      asc   "              ",00
```

```
    bra     §Reload      ; Et on réessaye la lecture...

No_MountErr cmp £1       ; Repositionne correctement la retenue
    _Err    "Impossible de charger le fichier musique : $"

    jsr     Erase_Playing
    lda     Playing_Music
    inc     Playing_Music
    asl
    asl
    tay
    lda     Parm_Music,y
    cmp     £$FFFF
    beq     End_Load
    brl     §More_Load

End_Load lda Playing_Music
    sta     NbMusic

    ldx     £190*$A0
    jsr     Print
    asc     "  ",0CFF
    asc     "CHANGER = ",0CEE
    asc     "ESPACE",0CFF
    asc     " / QUITTER = ",0CEE
    asc     "Q",0CFF
    asc     " ",00
                         ; Récupère adresse des pseudo-volumes
                         ; des tracks et des instruments
    pha
    pha
    ldx     £$12DB       ; _STGetVuPtr
    jsl     $E10000
    _Err

    pla
    pla
    tsc
    sec
    sbc     £4
    tcs
    phd
    inc
    tcd
    lda     $00          ; Et patche la routine de VUMètre
    sta     Patch_TrackVu ; pour  utiliser la table des tracks
    lda     $02          ; (Pour utiliser la table Instruments, il
    sta     Patch_TrackVu2 ; suffirait d'ajouter $20 à
                         ; l'adresse retournée)
    pld
    tsc
    clc
    adc     £4
    tcs

    ldx     £106*$A0+50
    jsr     Print
    asc     "TRACK VU-METRE.",00

    sep     $20

    ldal    $E1C023      ; Déclenchement de l'interruption SCB
    ora     £2
    stal    $E1C023
    stal    $E1C010
    lda     £$40
    stal    $E19DC7      ; Interruption SCB sur la 199ème ligne

    rep     $20

    lda     £1           ; On va commencer par la musique no1
    sta     Playing_Music

§NewMusic = *

* Sélection de la musique à jouer

    pea     $0           ; Numéro de la musique à sélectionner
Playing_Music = *-2
    ldx     £$0EDB       ; _STSelectBatch
    jsl     $E10000
    _Err

    jsr     Show_Playing ; Positionne ON JOUE en face de la
                         ; musique choisie...

* Lancement de la musique sélectionnée

    pea     $0           ; LoopOff
    ldx     £$0ADB
    jsl     $E10000      ; _STPlayMusic
    _Err

* Boucle Principale

§MainLoop pha
    ldx     £$0CDB
    jsl     $E10000      ; _STGetEofMusic
    pla
    oral    $E1BFFF
    bpl     §MainLoop    ; On sort de la boucle :
                         ; si : - une touche a été frappée.
                         ;      - la musique est finie.
    sep     $20
    ldal    $E1C000
    stal    $E1C010
    rep     $20
    and     £$00FF       ; Si aucune touche n'a été appuyée,
    cmp     £$0080       ; la musique vient de se terminer:
    bcc     ChangeMusic  ; on lance donc la suivante...

    cmp     £" "         ; ESPACE: on change de musique
    beq     ChangeMusic

    cmp     £"Q"         ; Q: Quitter
    bne     §MainLoop    ; Retour à la boucle principale
    beq     Quit_Mode

ChangeMusic jsr Erase_Playing
    lda     Playing_Music
    inc
    cmp     £4           ; On n'a chargé que 3 musiques
NbMusic=    *-2
    bne     No_DepMusic
    lda     £1
No_DepMusic sta Playing_Music
    brl     §NewMusic    ; Sélection et lancement de la
                         ; nouvelle musique

Show_Playing jsr Get_Adr
    jsr     Print
    hex     0CEE
    asc     "ON JOUE :",00
    rts

Print_Loading jsr Get_Adr
    jsr     Print
    hex     0CEE
    asc     "CHARGE : ",0CFF00
    rts

Erase_Playing jsr Get_Adr
    jsr     Print
    hex     0CEE
    asc     "        ",00
    rts
```

```
Get_Adr lda Playing_Music
        dec
        asl
        asl
        asl
        clc
        adc     £141            ; Numéro de ligne de la première musique
        asl                     ; (à remonter si on veut charger plus de
        asl                     ; musiques...)
        asl
        asl
        asl
        pha                     ; x $20
        asl
        asl                     ; x $80
        clc
        adc     1,s             ; x ( $80 + $20 ) = x $A0
        plx
        tax
        rts

Quit_Mode sep $30
§lp     ldal    $E1C02E
        cmp     £$F8
        bcc     §lp
        lda     £1              ; Passage en mode texte
        stal    $E1C029
        lda     £0
Save_C034 = *-1
        stal    $E1C034
        ldal    $E1C023
        and     £2|$FFFF
        stal    $E1C023         ; Arrêt interruption SCB
        rep     $30
        lda     £0
Save28 =        *-2
        stal    $E10028
        lda     £0
Save2A =        *-2
        stal    $E1002A         ; Remet le vecteur SCBInt
                                ; à sa valeur initiale
        ldx     £$03DB
        jsl     $E10000         ; _STShutDown
        _Err

        jsl     $E100A8         ; _Quit
        da      $29
        adrl    QuitRecGS

QuitRecGS da 0
        adrl    0
        da      0

* Exemple de VUMètre en Interruption Graphique...

TBAVU =         *
§A      =       0
        lup     32
        da      32-§A*$A0
§A      =       §A+1
        --^
        da      $2000

OldVU ds        $20

Inter phb
        phk
        plb

        rep     $30
        pei     $00
        pei     $02
        lda     £0              ; Adresse de la table de pseudo-volumes
Patch_TrackVu = *-2             ; retournée par l'outil
        sta     $00
        lda     £0
Patch_TrackVu2 = *-2
        sta     $02

        ldy     £VUNbTrack*2-2

§More_VuMetre lda OldVU,Y
        cmp     °$00§,Y
        beq     End_Update
        bcs     VuMetreDown

* VuMetreUp
        lsr
        lsr
        lsr
        pha
        lda     °$00§,Y
        lsr
        lsr
        lsr
        inc
        sta     New_Lgn
        pla
§loop   pha
        asl
        tax
        tya
        asl
        clc
        adc     TBAVU,X
        tax
        lda     £$1111
        stal    VULgn*$A0+VUCol+$E12000,X
        pla
        inc
        cmp     £0
New_Lgn =       *-2
        bcc     §loop
        bra     End_Update
VuMetreDown = *
        lsr
        lsr
        and     £%1111_1111_1111_1110
        tax
        tya
        asl
        clc
        adc     TBAVU,X
        tax
        lda     £$0000
        stal    VULgn*$A0+VUCol+$E12000,X
End_Update lda °$00§,Y
        sta     OldVU,Y
        sec
        sbc     £6
        bcs     NoDepVu
        lda     £0
NoDepVu sta °$00§,Y
        dey
        dey
        bpl     §More_VuMetre

        pla
        sta     $02
        pla
        sta     $00

        sep     $30             ; Fin d'interruption
        ldal    $E1C032
```

```
      and     £$DF
      stal    $E1C032
      plb
      clc
      rtl

* Routine d'affichage de caractères non proportionnels...
* Retour chariot  : caractère $8D
* Changement  couleur du texte : car. $0C + couleur (ex:$0CFF)

      mx      %00
Print pla
      sta     Msg
      jsr     Print0
      lda     Msg
      pha
      rts

Print_Rout= *
      dec
      sta     Msg
Print0 =      *
§lp2 stx      XValue
§lp  inc      Msg
      lda     !$0000
Msg =         *-2
      tay
      and     £$00FF
      beq     EndPrint
      cmp     £$0C
      beq     Change_Color
      cmp     £$8D
      beq     NewLine
      jsr     Print_Char
      bra     §lp
Change_Color tya
      xba
      sep     $20
      sta     Color
      rep     $20
      inc     Msg
      bra     §lp
NewLine lda £0
XValue =      *-2
      clc
      adc     £8*$A0
      tax
      bra     §lp2
EndPrint rts

Print_Char sec
      sbc     £' '
      asl
      asl
      asl
      clc
      adc     £Font
      sta     Font_Adr
      lda     £8
§lp2 pha
      phx
      lda     !$0000
Font_Adr =    *-2
      inc     Font_Adr
      ldy     £4
§lp  phy
      pha
      ldy     £0
      bit     £%0000_0010
      beq     No_Left
      ldy     £$0F
No_Left bit £%0000_0001
      beq     No_Right
      tya
      ora     £$F0
      tay
No_Right tya
      sep     $20
      and     £$FF
Color =       *-1
      stal    $E12000,x
      rep     $20
      inx
      pla
      lsr
      lsr
      ply
      dey
      bne     §lp
      pla
      clc
      adc     £$A0
      tax
      pla
      dec
      bne     §lp2
      txa
      sec
      sbc     £8*$A0-4
      tax
      rts

* Police de caractères  (voir le fichier sur le disque)

* Paramètres des musiques

Parm_Music da Txt_Music1,Music1
      da      Txt_Music2,Music2
      da      Txt_Music3,Music3
      da      $FFFF        ; Fin de liste

Music1 STR  '9/Music/Modulae'
Music2 STR '9/Music/Intro'
Music3 STR '9/Music/Toolbox'

Txt_Music1  asc " MODULAE MUSIC  (L.PICAMELOT)",00
Txt_Music2 asc " MODULAE INTRO  (S.MOUSSERON)",00
Txt_Music3 asc " TOOLBOX SONG   (L.PICAMELOT)",00

* Gestion des erreurs
      mx      %00
PgmDeath0 ldx £^Death_Msg
      ldy     £Death_Msg
PgmDeath pha
      phx
      phy
      sep     $20
      ldal    Save_C034
      stal    $E1C034
      rep     $20
      ldx     £$1503
      jsl     $E10000
Death_Msg str 'FATALE ERREUR SYSTEME : $'
```

## Le meilleur Apple.

- Quel est le meilleur ordinateur d'Apple?

- Pas l'Apple II GS. Pas le Macintosh. C'est la LaserWriter. Le coup de génie commercial d'Apple a été d'inverser les choses dans ses publicités, et de présenter le périphérique (un simple terminal clavier-écran de la Laser) comme si c'était l'unité centrale, sous le nom de "Macintosh". JYB

# 15 Octobre 1990: Apple relance l'Apple II !

J.Y. Bourdin

*15 Octobre 1990, 3614 Apple.* Curieuse politique commerciale que d'attendre trois semaines après l'Expo pour annoncer ses nouveautés. Bon, mais alors là, l'événement: trois nouveaux Macs, et... un nouvel Apple II ! Et quel Apple II ! Une rigolade comme ça, ça se partage...

### *Les nouveaux Macs: du bas de gamme ?*

- Le système livré avec ces Macs est le 6.06: riche en bugs, toujours pas de système 7. *Attendons.*
- Le Mac Plus/SE continue, sous le nom de Classic, mais désormais vendu au prix qu'il vaut. Les logiciels Mac coûtent plus cher que les Macs. A part ça, toujours très en-dessous du GS: monochrome, pas de slot, pas de RamDisk ou de RomDisk, ne boote pas sur Appletalk... Il paraît qu'on a caché dans sa Rom de 512k quelques amorces de cela pour le système 7. *Attendons.*
- Le Mac SI plane dans des sphères de prix comparables à celles du Baby-Next... *Il faudra comparer: attendons.*
- Le troisième, le LC, voudra être un Mac couleur bas de gamme (17 000F la version couleur: 'Low-Cost', tout est relatif). Ca ne sera pas un Mac II (donc ça ne sera pas grand-chose), et son unique slot sera compatible avec rien de ce qui existe. Le IIc Plus du Mac. *Il n'y en a pas chez le concessionnaire: c'est du Vaporware, attendons.*

### *Le nouvel Apple II: en arrière, toute !*

Le plus drôle, et de loin, le gag informatique de l'année, c'est la quatrième machine annoncée en ce mémorable 15 Octobre: sous la forme d'une carte qui bouchera l'unique slot du LC, Apple sortira en 91 **un nouvel Apple II**. Tenez-vous bien, ce sera... **un Apple 2E**!

Apple, je te remercie: ça fait un bout de temps que tu ne nous avais pas offert une rigolade pareille! Quel dommage, quand même, qu'Emile Schwarz n'ait pas pu nous montrer ça à l'Expo, ça l'aurait bien égayée! Demi-tour, coco, et à fond la caisse: horizon EuroPlus et Integer Basic, l'an 2000 s'annonce bien. *On rattrapera bientôt le ZX 81!*

Bon sang, une carte Apple IIe, c'est tout ce que tu sais faire, Apple? *Sera-ce une révision A? Aura-t-elle le Dos 3.2, ou chargera-t-elle Sweet 16 dans la carte langage par le port cassette?*
Quatre ans en retard derrière IBM (eh oui, cela fait quatre ans qu'une carte IIe existe sur IBM), chapeau pour l'avance technologique! Si Apple croit damer le pion de Big Blue auprès des écoles américaines avec cette course à reculons, il se trompe: il ne fait que le cautionner ("voyez, Apple s'y met aussi"). **Sur le terrain du retard technologique, IBM est imbattable.**
Et pourtant, voici Apple France, qui avait cessé la vente du 8 bits depuis un bon moment, qui avait refusé de vendre le fossile précédent appelé IIc Plus, le voici, donc, qui remet Emile Schwarz à l'Integer Basic!

Renseignements pris, il semblerait qu'on ne songe pas à nous fourguer ces vieilleries en échange de nos GS. Je l'ai déjà écrit: *au-dessus d'un GS, il n'y a que le Mac II.* Pour les "Downgrade", chercher d'autres gogos.
Je me demande bien quels indigènes pourront accepter sans rire les pièces de brocante d'Apple. Eh, Sculley, si ton stock t'encombre, je connais quelqu'un aux puces de Saint Ouen qui débarrassera ton grenier pour pas trop cher...

### *Mac II GS...*

Apple, à sa façon bien à lui, a l'air de se rendre compte qu'il existe des utilisateurs individuels, qu'il existe même des Apple II.

Certes, si les nouveaux Macs sont à la gamme Mac ce que le nouvel Apple II est à la gamme II, alors c'est **vraiment du bas de gamme**: du **très très bas** de gamme.

Mais enfin, cela devrait suffire pour qu'on cesse de nous bassiner sur la soit-disant mort de l'Apple II :

### *Apple n'abandonne pas l'Apple II: il se défonce sur le 8 bits!*

Laissons-le préparer un nouvel EuroPlus, et quand à nous, avec nos Apple II GS, faisons nos économies. Car pendant ce temps, ***le "Mac II GS" se prépare***: c'est la carte Duet.

# Compléments sur le Syquest (🍎)

Hubert Loiseleux

Je n'ai sans doute pas été suffisamment clair dans mon article sur le Syquest dans Toolbox-Mag N°1, si j'en juge à quelques discussions que j'ai pu avoir depuis.
La question de fond est *"Qu'est-ce que le Syquest?"*. Et la réponse n'est pas tout-à-fait 'un nouveau disque dur'; elle n'est pas non plus 'un périphérique pour les back-ups'. Voici cette réponse, telle que la donne la très officielle documentation technique du constructeur ("Syquest SQ555 Installation and Operation Guide", page VI):
*«Le SQ555 est un lecteur de disques (Disk Drive) 5,25 pouces Haute Performance, à cartouches de 44 Mégaoctets formatées, au standard SCSI»*. C'est donc simple: **c'est un lecteur de disques**, ces disques étant des cartouches d'une contenance égale à celle d'un disque dur.
Il est vrai qu'il y a eu une époque lointaine où les périphériques fabriqués par Syquest étaient destinés à faire fonction d'utilitaires de back-up, à remplacer les périphériques à bande (tape back-up).
Mais justement,si on trouve désormais des Syquest partout (brouette, casserole, Mac, GS, etc), c'est que Syquest commercialise une 'gamelle Syquest' qui est conçue au départ pour prendre la place d'un lecteur 5,25 sur les machines MS-Dos. C'est cette 'gamelle' qui est mise en boîte par différents constructeurs, et livrée pour différentes machines. C'est un lecteur de disques SCSI, point final.
Cela permet de régler quelques questions annexes:
1/ Quand vous aurez une carte Duet dans votre GS, vous serez fort heureux d'avoir seulement à changer la cartouche pour avoir votre disque dur en Mac comme en GS. Ce sera aussi simple que pour les disquettes 800k, sauf qu'il s'agira d'une disquette 42 Mégas.
2/ Certains parlent de 'problèmes de fiabilité' à propos du Syquest: entendons-nous. Le Syquest est un lecteur de disques à cartouche amovible. La fiabilité du Syquest ne peut donc pas se mesurer indépendamment de la qualité des cartouches qu'on y met: exactement comme pour les disquettes 800k, où la fiabilité dépend essentiellement de la qualité de la disquette.
Même avec une cartouche de qualité, la probabilité d'apparition de mauvais blocs ("bad blocks"), phénomène normal sur tous les disques durs, est évidemment plus grande sur un disque amovible que sur un disque non-amovible.
Si vous changez souvent de cartouche, les "bad blocks" apparaîtront plus vite que sur un disque dur non-amovible, c'est inévitable. Il faut dans ce cas accepter d'avoir à reformater de temps en temps la cartouche. Mais précisément, c'est sans importance, *puisque vous avez votre back-up sur une autre cartouche strictement identique à la première*! Bref, c'est comme avec les disquettes 800k, sauf qu'elles font 43 Mégas.
Sur un disque dur non-amovible, vous aurez de toute façon des mauvais blocs qui apparaîtront tôt ou tard. Simplement, dans ce cas-là, vous ne pourrez pas mettre l'autre cartouche à la place!
3/ Les cartouches Syquest peuvent être, comme tout disque, protégées contre l'écriture. Voilà une fonction simple et évidente, qui manque pourtant à beaucoup de disques durs.
4/ La cartouche du Syquest n'est pas *éjectable*, elle est *amovible*. Le changement de cartouche se fait à la main. Forcément: Syquest éteint ou allumé, il ne faut retirer la cartouche qu'une fois le moteur bien arrêté.
Pour le reste, j'avoue n'avoir rien compris à certaines discussions que j'ai pu entendre pour savoir si le Syquest était ou non "éjectable": je renvoie la balle à J.Y. Bourdin.

## Un mot de J.Y. Bourdin:

*"Homme libre, toujours tu chériras l'Apple II"*: tu as un GS, Hubert, cela ne permet guère de comprendre les angoisses des esclaves.
Les esclaves, ce sont les utilisateurs de Macintosh face à leurs disques. Tu n'ignores pas que quand tu appuies sur le bouton du second lecteur 3,5 d'un Mac (le même lecteur que le tien), il ne se passe rien du tout. Eh bien, c'est pareil avec le Syquest: si tu veux enlever une cartouche à la main, sans rien faire d'autre, ça ne marche pas. Il faut demander humblement la permission par soft à ton ordinateur: c'est lui le maître, l'utilisateur en est le serviteur.
Sur Mac, la question de savoir si le Syquest est "éjectable" est donc importante. Mais Prodos 8 et GS/OS sont conçus pour des hommes libres: le système se débrouille avec les disques qu'on lui met, c'est l'utilisateur qui décide.
J'ai un Syquest depuis plus de six mois, avec une carte SCSI Rév C, et je boote sur une carte Ram, le tout sous 5.02. Tout ça marche sans problème (sans ça, il n'y aurait pas de Toolbox-Mag!). Quand je suis sous le Finder, j'ai deux icônes pour le Syquest, correspondant aux deux partitions. Je n'ai jamais essayé d'envoyer ces deux icônes en même temps à la poubelle: ça ne servirait à rien, la cartouche devant de toute façon être enlevée à la main. Quand je retire la cartouche, les deux icônes passent en grisé, et quand je mets une nouvelle cartouche, ses icônes apparaissent. Bref, c'est strictement comme pour les disques 3,5 ou 5,25: **on fait ce qu'on veut, le système suit.** De toute façon, 99,99 % des utilisateurs de Syquest sur GS s'en serviront pour booter GS/OS. Et dans ce cas, en cas de changement de cartouche, il faudra de toute façon qu'ils rebootent, puisque GS/OS a besoin d'avoir le disque de boot en ligne.

## Un mot d'Yvan Kœnig:

Durant AppleExpo, plusieurs interlocuteurs ont formulé le souhait de pouvoir connecter à leur GS un lecteur 3,5 FDHD. Soyons sérieux, ce lecteur figure au tarif APPLE daté du 5 septembre avec un prix TTC de 4744 francs pour la version externe (avec boitier). Pour cette somme on peut se procurer un disque dur SCSI de 40 mégas. Or, la carte interface existe pour ce dernier alors qu'elle reste à élaborer pour le lecteur FDHD. Tirez en vous mêmes les conclusions qui s'imposent.

# Toolbox-Mag: Questions et réponses (*)

J.Y. Bourdin.

*L'Expo l'a montré, vous avez envie d'en savoir plus sur Toolbox-Mag que ce que peut en dire l'Edito du Directeur. Voici donc, à votre demande, quelques précisions sur le projet de Toolbox-Mag, présentées sous forme de questions-réponses.*

***- Savez-vous, Mr Bourdin, que vous nous avez déçus en cessant "Apple II for ever" dans Pom's?***

- Oui. Mais ce que j'ai écrit dans Pom's est exact. J'en avais assez d'écrire "Apple II for ever", même si cette chronique avait du succès (encore merci de vos compliments): il faut savoir s'arrêter à temps, avant le déclin (dans Toolbox-Mag, ce n'est pas moi qui assurerai la partie des News GS).
Mais surtout, il m'apparaît depuis un certain temps que l'urgence pour une revue française Apple n'est pas tant de rapporter fidèlement ce qui se passe aux USA, que de promouvoir et faire connaître le magnifique travail des programmeurs et développeurs français et européens pour le GS.
Je sais bien que beaucoup étaient prêts à continuer à lire un "Apple // for ever" sans changement dans un Pom's sans changement. Et Pom's était évidemment tout prêt, lui aussi, à cette continuité tranquille.
Seulement voilà: il y a un tournant à prendre, des changements importants à opérer pour qui veut défendre le GS aujourd'hui. Voilà la raison de fond qui fait qu'il n'y a plus d'"Apple // for ever", et qu'il y a Toolbox-Mag.

***- Des changements? Quels changements?***

- Si Toolbox-Mag est né, c'est que Toolbox a accepté, parce qu'il les partage, quelques grands principes qui définissent ce que j'appellerai la 'charte de Toolbox-Mag'. Ils sont au nombre de cinq.

1/ **Prendre au sérieux l'Apple II GS**, montrer sans préjugés ce dont cette machine est capable. Et cela sans sectarisme comme sans vaine querelle, en particulier à l'égard du Mac. La plupart des utilisateurs de GS n'utilisent leur machine qu'à 10% de ses capacités. La mission d'un magazine Apple n'est pas de leur vanter une autre machine, mais de leur montrer comment utiliser pleinement la leur. Si jamais un jour, ayant utilisé les 90% restants, ils butent sur les vraies limites du GS, ils sauront évoluer d'eux-mêmes.

2/ **Faire un magazine de bon niveau, qui respecte ses lecteurs**, qui essaie d'éviter la publication de trop grosses bêtises, qui s'engage de toute façon à rectifier systématiquement les erreurs qui lui auront été signalées, qui écarte les polémiques subalternes, les considérations personnelles oiseuses, etc. Mieux vaut publier des informations utiles et des programmes de qualité.
Un magazine qui **respecte également ses auteurs**, au minimum en accusant réception de leurs envois, mais aussi en acceptant de les informer du devenir de leur travail et de discuter avec eux des éventuels changements à y apporter.

3/ **Faire un magazine informatique moderne**, où le lecteur n'ait pas à taper une seule ligne, débarrassé des listings rebutants et des pleines pages de codes-objet hexadécimaux, mais publiant pourtant des programmes entiers: pour cela le magazine doit impérativement **comprendre la revue et la disquette, et concevoir les deux ensemble.**

4/ **Avoir une vraie politique rédactionnelle**, ce qui signifie ne pas se contenter de trier dans ce qu'on reçoit, mais aussi solliciter des auteurs possibles en fonction de ce que le magazine a décidé d'être, et de ce que ses lecteurs souhaitent et apprécient, et assurer un véritable **suivi rédactionnel** d'un numéro à l'autre.

5/ Constituer **le lieu naturel d'expression et de publication des programmeurs et développeurs français pour l'Apple II GS**, et avant tout de ceux qui ont le plus besoin de soutien: les plus jeunes d'entre eux. Ils font un travail magnifique, sans aucune aide ni soutien d'Apple-France, et sont souvent connus et reconnus aux Usa, y compris par Apple-Cupertino, mais méconnus en France. Il faut faire reconnaître la valeur de leur travail, et parfois aussi les guider un peu en ce qui concerne les exigences de la commercialisation et de la publication.

***- Toolbox-Mag est-il un magazine exclusivement Apple II GS?***

- Non. Il est destiné aux utilisateurs d'Apple II GS, parce qu'il prend l'Apple II GS au sérieux, et cela lui suffit pour trancher dans l'ensemble de la presse informatique française.
Mais les utilisateurs de GS ont l'esprit ouvert. Dès ce numéro, François Hermellin, que nous accueillons au Conseil de Rédaction, nous a concocté des "Exotica" que nous publions avec plaisir.
**Toolbox-Mag n'est lié à aucune marque**, n'a aucun fil à la patte avec Apple ou toute autre puissance économique, ne contient aucune publicité. Si demain arrive un chouette Baby-Next, si la carte Duet agrandit notre GS, ou tout autre événement qui passionnerait les utilisateurs de GS, Toolbox-Mag ne se ferme d'avance aucune porte.

***- J'ai un //c. Toolbox-Mag contiendra-t-il des programmes en huit bits?***

- Toolbox-Mag est a priori ouvert à tous les programmes tournant sur GS, 8 ou 16 bits. Mais attention, '8 bits' n'est pas du tout équivalent à 'tournant sur //c': **la compatibilité avec les Apple II du passé ne sera**

**en aucun cas un souci de Toolbox-Mag.** Avez-vous songé à regarder le prix d'occasion d'une unité centrale de GS?

***- Toolbox-Mag supportera-t-il le nouvel Apple IIe d'Apple, la carte à mettre dans un Mac?***

- Sur les Apple II du passé, voir ci-dessus.

***- Toolbox-Mag est-il destiné seulement aux programmeurs? Quel niveau faut-il pour le lire?***

- Quel que soit votre niveau de départ, la fonction de Toolbox-Mag est de vous permettre de progresser dans la maîtrise des possibilités de votre GS. Il est donc destiné aussi bien à l'utilisateur débutant qui utilisera les programmes, les fontes, les modèles, etc, de la disquette, qu'au programmeur chevronné qui concevra son propre Tool GS.
Nous ne demandons que deux choses au lecteur de Toolbox-Mag: d'abord, avoir lu les documentations qui lui ont été remises avec son GS. Ensuite, avoir envie de progresser, d'apprendre.

***- Quels logiciels dois-je avoir pour profiter pleinement des programmes Toolbox-Mag?***

- La plupart des programmes publiés par Toolbox-Mag ne demandent aucun logiciel particulier autre que la dernière version française du système Apple. Evidemment, si vous voulez modifier, réassembler, recompiler, etc les programmes-sources de la disquette, il faudra avoir les logiciels correspondants (APW, Merlin, Orca/C, TML Pascal, etc).
Nous avons besoin également d'un logiciel qui permette la communication entre nous. Nous avons choisi **AppleWorks-GS**, de Claris: non par amour pour Claris, mais parce que c'est le seul logiciel intégré en mode GS.

***- Toolbox-Mag est-il réservé à la programmation 'orthodoxe' de l'Apple II GS?***

- Non. Tous les modes de programmation et d'utilisation de l'Apple II GS ont leur place dans Toolbox-Mag. Nous sommes heureux d'accueillir à notre Conseil de Rédaction *Olivier Goguel*, et avec lui tout le joyeux gang du 'FTA' (Free Tools Association). *Photonix II*, *Modulæ* et les autres sont la preuve de ce qu'on peut faire en programmant le GS 'sur le métal'.
*Space Shark* est produit par un autre joyeux gang, et il fait lui aussi la preuve que si vous programmez votre GS sans intermédiaire, vous ne direz plus jamais qu'il est trop lent (pour les mollassons du joystick comme moi, ce jeu est en fait beaucoup trop rapide). *Claude Pélisson*, l'un de ses éminents auteurs, entre aussi dans notre Conseil de Rédaction!
A vous tous, programmeurs GS, qui avez fait de si belles choses, mais étiez bien forcés de diffuser en freeware ou en shareware dans le petit cercle des pseudos bien connus, nous disons: **Toolbox-Mag est votre magazine, c'est votre lieu naturel de publication.** Sortez au grand jour, à l'air libre, et montrez à tous ce que vous savez faire de votre GS. Un jour viendra du débarquement en force de ces "funny frenchies" Outre-Atlantique: n'aimeriez-vous pas être du voyage?

***- Y aura-t-il des logiciels en freeware et shareware sur les disquettes Toolbox-Mag, comme il y en a sur celles d'A2 Central, 8/16 ou Pom's?***

- Non. La diffusion de freeware et de shareware est un autre travail que la fabrication d'un magazine, et ce travail est fait par ailleurs.
Si certaines revues ont cette pratique, c'est qu'elles y sont obligées pour pouvoir remplir et vendre la disquette, laquelle est conçue comme un produit séparé, à part de la revue. La formule moderne de Toolbox-Mag (revue et disquette inséparables) lui permet d'éviter cette pratique fort suspecte: utiliser gratuitement le travail des autres comme rallonge à un produit incapable de se vendre seul.
De toute façon, dans un magazine 'G' et 'S', les graphiques et les sons prennent de la place. Il n'y a **pas de place sur la disquette pour le freeware** .
On pourra peut-être un jour ré-examiner la chose avec la disquette "bis", une nouveauté de ce numéro. Mais elle aussi est avant tout destinée à contenir ce qui déborde de Toolbox-Mag.
Tout le contenu de Toolbox-Mag (revues et disquettes) est et reste donc entièrement sous le Copyright impitoyable de Toolbox-Mag.

***- Pourquoi n'est-il pas possible d'acheter Toolbox-Mag au numéro?***

- La raison de fond est inscrite dans la "charte de Toolbox-Mag": ce magazine est doté d'un vrai suivi rédactionnel. Le N°2 comprend déjà des correctifs au N°1, et la suite d'une série commencée dans le N°1. Il en sera de même à chaque numéro.
De plus, cela simplifie considérablement la gestion, donc diminue les dépenses.

***- Comment faire alors pour avoir les anciens numéros?***

- Pour le moment, il n'y a pas d'anciens numéros, tous les abonnements commencent au numéro 1.

***- Quelle est la périodicité de Toolbox-Mag?***

- Toolbox-Mag n'a pas de périodicité absolue: il est possible que vous receviez un jour deux numéros d'un coup, qu'il y ait des numéros spéciaux, etc. L'abonnement couvre six numéros (six revues et six disquettes), et nous solliciterons votre réabonnement au bout de ces six numéros.
Ceci dit, nous suivons une périodicité indicative de deux mois: le premier numéro était daté d'Octobre 90 (parution légèrement anticipée pour l'Apple Expo fin Septembre), ce numéro-ci est daté de Décembre 90, le suivant sera daté de Février 91, etc.

***- Pourquoi le prix de l'abonnement est-il monté à 590F? Pourquoi certains de mes amis reçoivent-ils les premiers numéros gratuitement?***

- 590F, c'est le prix normal, l'abonnement ne descendra plus en-dessous de ce tarif. Mais il était normal d'avantager ceux qui ont aidé au lancement de Toolbox-Mag. Les membres de Toolbox avant Toolbox-Mag bénéficient de trois numéros gratuits. Les abonnés de Septembre ont bénéficié du tarif de lancement. Quand vous recevrez une proposition de réabonnement, n'attendez pas: profitez de l'occasion qui passe!

***- Il n'y a pas de publicité dans Toolbox-Mag, mais on dirait bien que Toolbox est un sponsor?***

- Toolbox n'est pas le sponsor, c'est *l'éditeur* de Toolbox-Mag. En retour, Toolbox vous envoie le catalogue de ses produits et peut vous faire ses propositions commerciales (ce qui commence par une réduction sur tous les produits Toolbox). Mais *Toolbox n'intervient en aucun cas dans le contenu de Toolbox-Mag*, pas plus qu'un éditeur n'intervient dans le contenu des livres qu'il édite.

***- Pourquoi le prix des produits n'est-il pas indiqué dans Toolbox Mag?***

- Toolbox-Mag est un magazine, ce n'est pas un catalogue commercial. Sa mission se borne à informer sur l'existence d'un produit et, éventuellemment, sur ses qualités et défauts.
Les prix sont choses fort variables, dépendants du vendeur et de la conjoncture. C'est le travail des commerçants d'informer leurs clients sur leurs produits. Nous renvoyons à eux.

***- Quel est le tirage de Toolbox-Mag? A combien fixez-vous le seuil de rentabilité? Quels sont vos objectifs financiers?***

- Les chiffres sont notre affaire, excusez-nous. Pour le moment, après le lancement (réussi), nous pouvons vous dire ceci: notre pari sera gagné si chacun d'entre vous abonne un ami. Toolbox-Mag ne fera pas de publicité dans la 'presse dévote', il ne peut compter que sur ses lecteurs comme agents publicitaires. Il n'a aucune recette publicitaire, et vit de ses abonnements. Le premier objectif financier de Toolbox-Mag, c'est d'augmenter dès que possible la rémunération de ses auteurs.

***- Je ne suis pas abonné à Toolbox-Mag, mais un ami me l'a prêté, et...***

- J'ai bien noté votre nom et votre adresse. Nous ne sommes pas tombés de la dernière pluie, notre siège social n'est pas Outre-Atlantique, et le monde est petit. L'encadré de la page 3 prévient les copieurs. Quand ça tombera, ne venez pas vous plaindre...

## *Wanted: suite*

❑ Evidemment, ne faîtes plus le Tool pour les musiques SoundSmith: c'est fait.

❑ En revanche, les compléments à cet outil nous seraient utiles. Nous serions preneurs aussi d'un morceau de TML Pascal à couper-coller, qui permettrait à n'importe quelle application de charger et jouer des musiques SoundSmith.

❑ Traduction en Merlin du StartupTools de Desnoues: c'est fait, merci Yvan Kœnig.

❑ Une chose à rajouter aux "Wanted", un mot magique: PostScript, le langage d'Adobe qui est dans la LaserWriter. Il y restera encore un bon moment, Apple a appris à le reconnaître comme standard de fait. Or, le GS ne sait que bien peu parler en PostScript à la Laser...

Donnons tout de suite une référence: il s'appelle *Don Lancaster*, c'est un bidouilleur fou sur l'Apple II. C'est aussi un expert incontesté du PostScript. Sa théorie: connectez la Laser sur le port série, et parlez-lui avec... AppleWriter et WPL!

Il existe également un programme en shareware britannique, dont nous vous parlerons dans un prochain Toolbox-Mag, conçu comme un langage pour le shell APW/Orca, qui permet de parler en PostScript.

❑ Wanted aussi, des XCmds, des commandes externes, pour HyperStudio bien sûr, mais aussi pour tout ce qui supporte des commandes externes: les shells comme APW/Orca, mais aussi Prosel et Merlin, Nifty List, etc. Une limite cependant: par définition, ces programmes exigent que l'utilisateur dispose du logiciel correspondant.

❑ Evidemment, la première XCmd à HyperStudio sera une commande pour jouer les musiques Music Studio et SoundSmith avec les outils Toolbox-Mag. La conception d'HyperStudio en matière de son ("Digitalisez") est stupide: elle mène à saturer la mémoire et les disquettes avec des musiques courtes et faibles. L'Apple II GS est quand même autre chose qu'un lecteur-enregistreur!

JYB

## Entendu à l'Expo

"Comment appelez-vous ce soit-disant logiciel Claris qui ne communiquerait pas ses fichiers avec Mac Write et Mac Draw? Appleworks-GS?

Ecoutez-moi bien, cher Monsieur. Vous êtes ici sur le stand officiel Claris à l'Apple Expo, et nous sommes les porte-parole autorisés de Claris-France. Je suis formel: si ce logiciel dont vous parlez, et que je ne connais pas, existe réellement, eh bien ce n'est pas un logiciel Claris."

# Initiation aux ressources, Première partie: Le Resource Manager (🍎🍎)

Bernard Fournier

Avec le système GS/OS 5.0 est apparue dans le monde du GS une nouvelle fonctionnalité: les ressources. Cette nouvelle forme de stockage et d'exploitation des informations se fait à l'aide du *Resource Manager*, l'outil numéro 30.

## Pourquoi des ressources ?

Jusqu'à présent, il n'y avait qu'une possibilité de conserver les informations sur disquette: on créait un fichier conservant les données. Ces fichiers pouvaient être des applications (les programmes) ou des data (textes, images, fontes, sons...). Lorsqu'un programme avait besoin de données complémentaires, il les chargeait à partir du disque. Si un programme avait besoin de plusieurs images, sons ou autres données, il fallait que chaque fichier de data soit présent sur la disquette. Inconvénient majeur: l'installation de telles applications sur disque dur nécessitait de bien faire attention à recopier scrupuleusement *tous* les fichiers de data dans un dossier, et surtout sans modifier leurs noms.

D'autre part, pour le programmeur, un certain nombre de fonctionnalités de la boite à outils étaient lourdes et complexes à mettre en oeuvre. Elles demandaient une part importante de programmation, ce qui alourdissait inutilement le travail. Par exemple, pour définir une fenêtre, il fallait écrire un code important; ou la procédure de chargement des outils: une bonne centaine de lignes dans le source, au bas mot, pour effectuer cette tâche indispensable. Et la gestion des erreurs... une application ayant besoin de charger et/ou sauver des données sur disque devait avoir prévu *tous* les cas d'erreurs susceptibles de se produire et devait, pour chacun d'entre eux, afficher un message d'alerte. Surcroit de travail, encombrement de la mémoire, risques d'oublis: tels étaient les pièges de la programmation avant l'utilisation du Resource Manager. Et je ne parle pas des dialogues standards de chargement/sauvegarde qui gardaient leurs messages en Anglais (les contenus des boutons OPEN, SAVE, NEW FOLDER...), ce qui était du plus mauvais effet dans un programme où tous les autres messages étaient en Français.

Désormais tout cela est un mauvais souvenir. Avec les ressources, plus besoin de fichiers de data: toutes les informations et données complémentaires sont en ressources, l'application n'a pas besoin de fichiers de data. La programmation est allégée: pour charger les outils, un simple StartupTools suffit. La définition d'une fenêtre? un appel à NewWindow2 et le tour est joué. Les messages d'erreurs? le Resource Manager s'en charge tout seul. Les contenus des boutons des dialogues standards? ils sont en français si on utilise GS/OS 5.02 francisé. Bref: les ressources, c'est l'assurance d'un code compact, d'une programmation claire et simplifiée et du confort pour l'utilisateur.

Et il y a un autre avantage... les données stockées sous forme de ressources sont éditables et modifiables à loisir par tout un chacun. Ce qui signifie que si une application est écrite par un Esquimau, il aura prévu des messages d'alerte, des menus et autres informations affichées à l'écran dans sa langue natale. Avant GS/OS 5.0, une seule solution pour franciser cette application: un éditeur de blocs et on change tous les messages (avec les incontournables problèmes de taille: trouver un équivalent français à PRINT en 5 lettres...). Tandis qu'avec les ressources, tout est simple: avec un Editeur de ressources on change la chaîne PRINT par IMPRIMER MON DOCUMENT et désormais le message s'affichera en français, dans le menu ou dans un dialogue, sans problème. Ainsi la francisation d'un programme écrit en Esquimau se réduira à un simple problème de traduction...

Comment cela est-il possible? prenons un exemple: pour afficher une chaîne de caractères comme PRINT, on procédait comme suit avant les ressources.

- on définissait une chaîne de longueur fixe (ici 5 caractères) contenant PRINT
- on faisait afficher la chaîne dans le menu FILE par exemple
- si on voulait franciser ce message, on ne disposait donc que des 5 caractères prévus à cet effet par le programmeur... d'où le casse-tête pour trouver un équivalent dans notre langue en si peu de lettres!

Mais avec les ressources, voici comment on procède désormais:

- on crée un menu pointant sur des chaînes de longueur variable
- on définit les chaînes
- on fait afficher le menu: celui-ci va afficher le contenu de la chaîne, quelle que soit sa longueur! donc si on remplace PRINT par IMPRIMER, cela ne perturbera aucunement notre application.

Ceci était un exemple simple, mais avec les ressources, tout est possible (en respectant un certain nombre de règles, bien entendu).

## Editer les ressources

Avant de plonger plus avant dans l'étude des ressources et de ce qui régit leur genèse et leur emploi, nous allons d'abord voir comment se présentent les ressources.

Les ressources sont stockées indépendamment du programme (le code), mais à sa suite. C'est à dire qu'on peut modifier les ressources d'une application sans rien changer au programme, et ces ressources seront toujours partie intégrante de l'application. En effet, une application contenant des ressources se présente sous la forme d'un fichier unique contenant deux types de données: le code exécutable et les ressources. Et c'est GS/OS et le Resource Manager qui font la différence entre ces types de données lors de l'exécution du programme.

Tout à l'heure, nous parlions d'éditer les ressources (pour les

### Faux ami

"Les ressources du Resource Manager": il y a de quoi mettre en défaut tous les correcteurs! Excusez-nous si nous avons laissé passer des erreurs...

## Type et Identifiant

Une application reconnaît les ressources à l'aide de deux paramètres: le *type* et l'*identifiant*. Le type définit un groupe de ressources de même nature: par exemple toutes les chaînes Pascal (type $8006), les fenêtres (type $800E)... alors que l'identifiant permet de différencier les ressources de même type. Par exemple, deux chaînes Pascal auront un type identique ($8006) mais se distingueront par leurs identifiants uniques: $1 et $2 par exemple. On pourra avoir des identifiants identiques dans des types différents: par exemple chaîne Pascal type $8006; id $1 et fenêtre type $800E; id $1.

En résumé: une ressource donnée doit être définie de manière unique par son type et son identifiant.

Le Resource Manager accepte les types suivants:
- $0000 : valeur interdite
- $0001 à $7FFF: valeurs disponibles pour le programmeur
- $8000 à $FFFF: valeurs réservées

Le Resource Manager accepte les identifiants suivants:
- $00000000: valeur interdite
- $00000001 à $07FEFFFF: valeurs disponibles pour le programmeur
- $07FF0000 à $7FFFFFF: valeurs réservées
- $08000000 à $FFFFFFFF: valeurs interdites

consulter ou les modifier). Ceci ne peut se faire qu'à l'aide d'un Editeur de ressources.

A l'heure actuelle, il n'en existe que trois:
- TML PASCAL // (COMPLETE PASCAL)
- GENESYS
- APW REZ et APW DEREZ

Note: *Design Master* est un générateur de ressources, et non un éditeur! C'est à dire que l'on peut créer ses ressources, mais qu'on ne peut pas modifier celles déjà existantes... fâcheuse lacune.

*L'éditeur TML*: c'est le plus simple de tous, mais aussi le moins complet (on ne peut pas éditer les codes de mise en page ni les ressources personnelles par exemple). Toutefois, pour un débutant, il permet de comprendre comment fonctionnent les ressources et également de les modifier aisément.

*L'éditeur Genesys*: très complet, puissant, performant, il permet toutes les manipulations possibles sur toutes les ressources standard (les ressources personnelles peuvent juste être consultées). Outre cette limite, on notera qu'il ne permet pas la création (mais la consultation est possible) des rPicture. Particularité: Genesys permet de transformer les ressources d'une application en source compilable par APW REZ. A mon avis, Genesys est indispensable, à plus d'un titre, pour les programmeurs se lançant dans les arcanes des ressources.

*L'éditeur APW REZ-DEREZ*: ces deux modules de l'environnement de programmation APW permettent de compiler des ressources et de les désassembler, bien entendu en mode texte (alors que TML et Genesys fonctionnent en mode 'desktop'). Le programmeur désirant tirer pleinement partie des ressources doit impérativement utiliser REZ et DEREZ. En effet, ce sont les seuls (à ce jour) permettant *toutes* les manipulations possibles sur *toutes* les ressources (ressources système *et* ressources personnelles). D'autre part, la création de ressources avec APW.REZ oblige le programmeur à bien connaître la structure des ressources; démarche trop souvent négligée par les inconditionnels de Genesys. Enfin, je signale que les personnes voulant utiliser des codes de mise en page dans des textes n'ont que la possibilité d'utiliser APW.REZ.

Pour ce premier article sur les ressources, nous allons voir comment transformer une ressource existante en fichier source compilable. Ceci nous permettra de mieux comprendre comment est constituée une ressource afin de pouvoir ultérieurement définir les nôtres. Ceux que seule intéresse la modification de ressources (par exemple pour franciser des messages) ont juste besoin de lire attentivement la documentation fournie avec TML Pascal ou Genesys.

## Désassemblage de ressources avec Genesys:

On édite le fichier ressource avec OPEN, puis on sauve le source correspondant en fichier .SRC (choisir le langage approprié dans CHOOSE LANGUAGE et prendre APW REZ) sans oublier de générer le fichier des EQUATES (options GENERATE SOURCE et GENERATE EQUATES). Voir la documentation du logiciel pour de plus amples renseignements.

Attention: les librairies de désassemblage fournies actuellement avec Genesys comportent certaines erreurs. En conséquence, les sources .SRC produits par cette méthode comportent des erreurs interdisant la re-compilation. Ces erreurs sont toutefois peu nombreuses et aisément détectables. Lorsqu'on va re-compiler les sources avec APW REZ, des erreurs de compilation seront signalées: il suffit alors de lister la ressource incriminée et en consultant le Tome 3 de la ToolBox Reference, on peut arriver à corriger la ou les erreurs (cependant, pour un débutant, cette tâche est un peu ardue... et donc déconseillée tant que les ressources gardent encore une aura mystérieuse pour lui).

Pour recompiler ce source, on utilisera APW REZ. Dans ce but, on réunira les deux fichiers: FIC.SRC et FIC.EQU en un seul source (avec les commandes copier-coller de APW) et on modifiera la directive #INCLUDE "TYPES.REZ" en spécifiant le préfixe complet d'accès à la librairie TYPES.REZ (on ajoutera également la directive #INCLUDE d'accès à la librairie TYPES.REZ.AUX si on la possède). Par exemple, on mettra des directives du style:

*#include "/HD/APW/R.LIBRARIES/TYPES.REZ"*
*#include "/HD/APW/R.LIBRARIES/TYPES.REZ.AUX"*
*... les equates*
*... le source des ressources*

## Désassemblage des ressources avec APW DEREZ:

Il faut tout d'abord posséder l'environnement de programmation APW et avoir installé le langage REZ dans le dossier LANGUAGES. Installer également l'utilitaire DEREZ dans le dossier UTILITIES et les librairies TYPES.REZ et PICT.REZ dans le dossier R.LIBRARIES. Ne pas oublier de mettre à jour le fichier SYSCMND (ajouter une ligne comportant DEREZ en première colonne

et *U dans la deuxième colonne; ajouter également une ligne avec REZ en première colonne et avec *L en deuxième colonne et 21 en troisième colonne).
La librairie TYPES.REZ étant incomplète, il est conseillé d'ajouter TYPES.REZ.AUX (qui se trouve sur la disquette Toolbox-Mag 2) dans le dossier R.LIBRARIES.
*Note*: il existe deux versions de REZ. La première, distribuée depuis Avril 90, comporte quelques erreurs (notamment lors des compilations de ressources volumineuses). Depuis Octobre, une nouvelle version est fournie par APDA; cette nouvelle version ne comporte plus ces erreurs.

## Pour désassembler un fichier ressources:

*DEREZ fichier.org -o fichier.dest types.lib*
(attention aux espaces !!!!!)
avec les conventions suivantes:
*fichier.org*: préfixe complet d'accès au fichier ressource à désassembler
*fichier.dest*: préfixe complet d'accès au fichier source généré qui sera conservé sur disque
*types.lib*: préfixe complet d'accès à la librairie Types.Rez
par exemple on aura une commande du style:
*DEREZ .D2/MonFich -o /HD/SRC/MonSrc /APW/R.LIBRARIES/Types.Rez*
Le fichier ainsi généré sera le source compilable ultérieurement par APW.REZ.
On va ensuite ajouter les directives #include d'accès aux librairies Types.Rez et TypeRez.Aux:
mettre en tête du fichier source les directives suivantes:
*#include "/HD/APW/R.LIBRARIES/TYPES.REZ"*
*#include "/HD/APW/R.LIBRARIES/TYPES.REZ.AUX"*
et voilà. Listez le source ainsi produit et muni du Tome 3 de la ToolBox Reference, essayez de décortiquer les ressources !
*Note*: les sources produits par DEREZ sont quelque peu 'touffus' et donc peu aisément compréhensibles. Ce que je conseille, c'est le désassemblage à l'aide de Genesys: ensuite on traque les -rares- erreurs que Genesys à produites.
La compilation à l'aide de REZ se fait de la manière suivante:
*REZ*
*COMPILE fichier.org keep=fichier.dest*
(attention aux espaces !!!!!)
avec les conventions suivantes:
*fichier.org*: préfixe complet d'accès au fichier source à compiler
*ficher.dest*: préfixe complet d'accès au fichier ressource généré qui sera conservé sur disque
A titre d'exercice, nous allons désassembler puis re-compiler les ressources du programme GS.PIC.MASTER situé sur la disquette Toolbox-Mag 2. Nous pourrons ensuite comparer les résultats avec le fichier source original LoadSavePic.S situé également sur cette même disquette.
Il est recommandé de travailler avec des copies de la diquette originale Toolbox-Mag 2!
*Désassemblage avec Genesys:*
- lancer Genesys
- faire OPEN et charger GS.PIC.MASTER
- indiquer le langage APW REZ dans Choose Language
- sauver le Source et les Equates
- lister et étudier le source. Le comparer avec LoadSavePic.S et Pic.Src fournis sur la disquette Toolbox Mag 2.
*-Compilation:*
passer en environnement APW et voir le paragraphe suivant pour la compilation (et les conventions employées).
*Note*: la librairie de décompilation de Genesys contient quelques erreurs; et donc le source produit ne peut être directement recompilé. Dans l'exemple qui nous intéresse, voici les corrections à apporter:
- remplacer RESOURCE $0C02 par DATA rImageAPF
- remplacer RESOURCE $C800 par DATA rMyFont (2 occurrences)
- remplacer R4F72_00002000 par CTLTMP_00002000
- ajouter en début des EQUATES:
*#Define rImageAPF $C02*
*#Define rMyFont $C800*

## Désassemblage/compilation avec APW:

(on suppose que REZ, DEREZ et les librairies TYPES.REZ et TYPES.REZ.AUX sont installés correctement dans votre environnement de programmation). Dans tout ce qui suit, on suppose que l'environnement de programmation APW se trouve sur un périphérique en ligne nommé /APW et que le fichier à désassembler est sur un périphérique nommé /B. Pour ceux ne disposant que d'un seul lecteur de disquette, recopier GS.PIC.MASTER sur la disquette /APW (s'assurer qu'il reste au moins la place pour archiver le source produit) et remplacer /B par /APW dans ce qui suit.
- désassemblage:
*DEREZ /B/GS.PIC.MASTER -o /B/LSP.Derez /APW/R.LIBRARIES/TYPES.REZ*
- édition et ajout des directives #include:
*EDIT /B/LSP.Derez*
passer en mode insertion (COMMAND-E) et ajouter en début du source les directives:
*#Include "/APW/R.LIBRARIES/TYPES.REZ"*
*#Include "/APW/R.LIBRARIES/TYPES.REZ.AUX"*
- sauver le source modifié
- lister et étudier le source. Le comparer avec LoadSavePic.S et Pic.Src fournis sur la disquette Toolbox Mag 2.
- compilation:
*REZ*
*COMPILE /B/LSP.Derez keep=/B/LSP.R*
*Note*: si le fichier source est trop volumineux, il ne pourra pas être édité par APW. Dans ce cas on le scinde en plusieurs parties avec l'utilitaire SPLIT (commande: SPLIT fichier.org size=500 basename=fich.dest); ou alors on utilise un traitement de texte, et on sauve le source en plusieurs parties en format TXT. Dans tous les cas, on ajoutera en fin de chaque source la directive #Append "nom de la partie suivante".

Etudiez attentivement le source LoadSavePic.S situé sur la disquette Toolbox Mag 2. Pour cela, il faut également lire le tome 3 de la ToolBox Reference (appendice E) afin de comprendre la constitution des ressources.
Dans Toolbox-Mag 3, nous étudierons la création de ressources ex-nihilo.

*Revue soft:*

# Tarot, de François Uhrich

par Laurent Bourdin

## Un jeu facile

Le jeu de tarot de François Uhrich (Toolbox éditeur) est un outil essentiel pour tout "taroteur". Que ceux qui ne connaissent pas le tarot sachent qu'il s'agit d'un jeu passionnant et dont les règles sont assez faciles, d'un jeu populaire.
Bien sûr, en lisant la documentation du Tarot, ils tomberont sur des phrases comme: "Si un contrat a été annoncé et si celui-ci n'est pas une Garde Sans ou Contre, le Chien est alors montré à tous les joueurs".
Mais précisément, le Tarot de François Uhrich est un fantastique outil pour apprendre le tarot: des règles claires sont disponibles dans le menu , les possibilités du logiciel permettront d'apprendre sans perdre sa chemise.
Par ailleurs, dans les différentes phases du jeu, de multiples aides guideront le novice (Aide à l'enchère, Aide a l'écart, Aide au jeu de la carte); n'en abusez pas trop cependant, car, à la limite, vous faîtes jouer le GS à votre place!

## Jouer avec le GS

Au départ, le temps de chargement du programme est un peu long (sur mon "petit" GS, 1,2 Mégas, dur 20 Mégas, pas de TransWarp, cela prend 57 secondes): mais ensuite, il n'y a plus d'accès disque du tout.
Le bureau habituel du GS apparaît (ce qui n'est pas si fréquent pour les jeux GS). Il suffit de demander NOUVEAU JEU dans le menu FICHIER pour pouvoir commencer à jouer. Ici se termine le GS, et commence le tarot.
C'est d'abord le traditionnel choix de carte pour désigner le donneur. Vient ensuite la coupe: le joueur désigné coupe avec un curseur transformé pour l'occasion en couteau!
A tour de rôle, en commencant par la droite du donneur, les joueurs (Nord, Est, Sud, Ouest, mais on peut modifier leurs noms) font leur annonce. Le jeu proprement dit commence.
Le chien s'affiche donc. Si le joueur a "pris" ou "gardé", un affichage lui permet de faire son écart. Le "jeu de la carte" commence alors.

A partir de là, il n'y a plus rien à dire: on joue au tarot, tout simplement. Le GS remplacera éventuellement de un à trois des autres joueurs.

## Des possibilités avancées

Le logiciel est doté de nombreuses possibilités au-delà du jeu simple:

♣ sauver un tournoi en cours de partie pour le recharger plus tard (quand il faut partir au bahut, c'est très pratique).
♣ jouer chacun son tour la même donne ou se mesurer à l'ordinateur (j'ai toujours perdu...).
♣ imprimer donnes et partie.
♣ créer soi-même ses donnes, les jouer, les sauver et les réutiliser.
♣ jouer à quatre: chaque personne est devant l'ordinateur tour à tour.
♣ voir le score
♣ voir la partie
♣ voir les statistiques: cette option permet aux joueurs en cours de partie de savoir non seulement quelles cartes sont déjà tombées, mais aussi qui possède (ou peut posséder) telle ou telle carte, qui coupe telle ou telle couleur..., le tout selon un savant calcul fait (comment?) par l'ordinateur.
♣ voir le dernier pli
♣ voir la donne en fin de partie

## Mon avis

Un choix du programmeur peut à mon avis se discuter: le logiciel se veut, et est, tellement beau à voir que cela détourne un peu du jeu lui-même, et ralentit un peu le jeu. Mais c'est un choix. Quoi qu'il en soit, j'ai particulièrement apprécié:

♥ le fait tout simple que ce logiciel soit français et en français, et permette de jouer à un jeu, le tarot, qui est bien de chez nous.
♥ la simplicité et la clarté du jeu et de son interface utilisateur: beau et clair, mais jamais bébête, comme le GS lui-même.
♥ l'humour toujours présent (les sons par exemple: quand le petit est perdu par une des parties, on peut entendre un magnifique AHRGG...)
♥ la beauté des cartes de J.F. SAUVAGE.
♥ l'absence de tout accès disque après le chargement, même avec un GS 1,2 Mégas.
♥ l'absence de toute protection disque, qui permet de mettre le jeu sur le disque dur.
♥ et puis surtout, c'est un tarot (un vrai, conforme au réglement fédéral et tout ça). Et j'aime le tarot.

# GS PicMaster (🍎🍎)

Bernard Fournier

## Une première mondiale (🍎)

Utilisateurs de GS, PicMaster GS est le seul logiciel au monde permettant d'afficher les images GS de n'importe quel format et de n'importe quelle taille: d'une fraction d'écran à plusieurs centaines d'écrans (avec possibilité de visionner l'image entière en réduction sur un écran)! C'est donc un programme indispensable.
Programmeurs, vous trouverez le listing complet des routines de sauvegarde et de décompactage pour tous les formats d'images: cela aussi, c'est unique... Et en prime, des modules bien pratiques: utilisation de fontes en Ressources, sauvegarde de Ressources, Custom Standard File.
NB: la routine Custom Standard File a été écrite par Carole Fox à partir d'une ébauche que je lui avais fournie en TML 1.5.

## 1 - Mode d'emploi (🍎)

PicMaster GS permet de charger en mémoire des images issues de n'importe quel format de fichier, puis de les sauver dans le format de votre choix, donc de les convertir.
Lorsqu'une image est chargée, le programme détermine automatiquement la palette à utiliser ainsi que la résolution (sauf pour les images écran qui ne conservent pas d'informations sur les SCBs). Lors de la visualisation d'une image, on peut changer la résolution (320 ou 640) en tapant sur la touche ESPACE.
Si l'image en mémoire occupe plus d'un écran, il est possible de faire défiler l'image avec les flèches (déplacement de 1 à 9 points à la fois si on tape sur les touches 1...9 et déplacement page par page si on tape sur 0). On revient au menu principal en cliquant la souris ou en tapant ESC.
Si l'image affichée occupe plusieurs écrans, on peut avoir une vision globale de l'image entière en tapant RETURN (on revient à l'affichage classique en cliquant).
Pour sauver une image, un dialogue Standard File permet de choisir le format de sauvegarde: PNT, SCR, APF ou SPF.
Lorsqu'une image est en mémoire, on peut également définir un rectangle (taille maximum = taille de l'écran): toute la partie sélectionnée peut alors être sauvée sous forme de Ressource (type $C02). Cette Ressource peut être ajoutée aux ressources d'une application et/ou archivée sur disque sous forme de source compilable par APW REZ.
Pour cela, choisir SAUVER RESOURCE: un dialogue de sauvegarde permet de spécifier le nom du fichier Ressource et son type.
- pour les fichiers Ressource compilés: le suffixe .REZ est proposé à la fin du nom (mais vous pouvez l'ôter). Si vous voulez ajouter la ressource dans un fichier Ressource, indiquez le nom du fichier et cliquer 'Remplacer' dans le dialogue d'alerte: la ressource va alors être ajoutée aux ressources existantes (si le nom indiqué n'est pas celui d'un fichier, PicMaster GS va créer un nouveau fichier Ressource sur disque).
- pour les fichiers Ressource sources: le suffixe .SRC sera ajouté d'office au nom que vous aurez spécifié et si un autre fichier existe déjà avec le même nom, la sauvegarde ne se fera pas.

### Avertissement

Le but de cet article est de décortiquer les routines PackBytes et UnpackBytes employées pour la sauvegarde des images GS. Les sources accompagnant le programme ont donc été écrits le plus clairement possible afin de permettre aisément la récupération des routines. Il est évident que le programme aurait pu être écrit de manière plus concise, mais au détriment de la lisibilité.
D'autre part, j'ai opté pour l'affichage direct en page $E1 au lieu d'utiliser le WindowManager. Ce choix délibéré est motivé par le désir d'expliquer le plus simplement possible le transfert d'images lors des chargements/sauvegardes. Dans un prochain article, nous aborderons l'affichage avec le Window Manager grâce aux structures LocInfo.

## 2- Les fichiers images GS (🍎)

Les formats de sauvegarde des images GS ont été définis par Apple afin de standardiser un maximum les choses. Les développeurs ont plus ou moins suivi ces indications, et on se trouve avec une situation assez embrouillée. Voici les formats définis par Apple:
- *fichier SCR*: écran sauvegardé point par point, type $C1
- *fichier PNT*: une palette et les motifs sauvés, puis compactage des points avec l'outil PackBytes, type $C0 auxtype $0
- *fichier SPF*: les points de l'écran et la palette utilisée compactés avec PackBytes, type $C0 auxtype $1
- *fichier APF*: les palettes compactées, les points compactés puis les motifs compactés (compactage avec PackBytes), type $C0 auxtype $2

Et maintenant les anachronismes:
- *fichier PNT*: GS Paint sauve ses images à ce format avec un auxtype $8000 pour signifier que la palette utilisée est la palette standard. Il faut donc indiquer que l'auxtype $8000 est valide pour la reconnaissance de ces images par le StandardFile dans vos applications
- *fichier SPF*: curieusement, personne jusqu'à présent ne s'est intéressé à ce format de fichier. Aucun logiciel commercial ne reconnait ce format (bien que The Graphic Exchange de R. Wagner soit capable de les afficher). Il existe à ma connaissance un seul logiciel de jeu utilisant des images de ce type: Dark Castle. Le programme PicMaster GS est donc un utilitaire bien pratique (et économique) pour visionner ces images.
- *fichier APF*: Apple ayant précisé que cette structure était évolutive, certains (Electronic Arts avec Deluxe Paint par exemple) ont utilisé cette possibilité pour sauver avec l'image des informations sur la position à l'écran.

Si on s'en tient aux spécifications Apple, il faut d'abord rechercher la chaine MAIN dans la suite d'octets pour déterminer le début de l'image proprement dite. Malgré ces recommandations, aucun programmeur n'effectue cette

## Lexique

### Les ScanControlBytes (SCBs)

Sur une page écran affichée (adresses $E12000 à $E19CFF) se trouvent 200 lignes de 160 octets. Chaque octet correspond à 2 ou 4 pixels (les points lumineux: on a donc une résolution horizontale de 320 ou 640 points). Chaque ligne est affichée en référence à une palette de couleurs (voir ce mot): chaque ligne peut donc afficher 16 couleurs à choisir dans la palette utilisée. Afin d'indiquer au GS comment afficher les couleurs de chaque ligne, une table contient, pour chaque ligne, l'indication de la palette retenue et du mode de résolution. Cette table s'appelle la *Table des SCBs*. En mémoire, cette table se trouve aux adresses $E19D00...$E19DFF (200 octets de SCB et 55 octets inutilisés). Chaque SCB est en fait un pointeur sur la palette utilisée ainsi qu'un indicateur de résolution.
L'octet correspondant se divise en deux nibbles de 4 bits:
Le nibble bas (bits 0...3) détermine la palette utilisée (de 0 à 15, suivant les 16 valeurs qu'il peut prendre)
Le nibble haut se décompose ainsi:

| | |
|---|---|
| bit 4 | toujours à 0 |
| bit 5 | mis à 1 -> FillMode actif (uniquement en mode 320) |
| | mis à 0 -> FillMode inactif |
| bit 6 | mis à 1 -> autorise les interruptions |
| | mis à 0 -> pas d'interruption |
| bit 7 | mis à 1 -> mode 640 points |
| | mis à 0 -> mode 320 points |

### Le fillmode

Lorsque le bit 5 du SCB est à 1 (en mode 320), le fillmode est actif. C'est à dire qu'à partir d'un octet de la ligne, tous les octets suivants mis à 0 sont affichés dans la même couleur.
Intérêt: en affichant seulement quelques points sur une ligne, on colorie toute la ligne instantanément (d'où vitesse accrue pour les animations !).
Exemple: *RoooooooooBoooVoooooooooRooo*
(R: point rouge B: point bleu V: point vert o: 'noir')
l'affichage se fera comme suit:
*RrrrrrrrrrBbbbbVvvvvvvvvvRrrr*

### Les interruptions

Lorsque le balayage de l'écran s'effectue, une interruption est générée si le SCB de la ligne en cours a son bit 6 mis à 1.

### Le mode

*mode 320:* chaque octet représente deux points lumineux (un point est représenté par 4 bits). Comme 4 bits peuvent donner 16 valeurs, on peut représenter chaque point par une des 16 couleurs de la palette.
*mode 640:* chaque octet représente quatre points lumineux (un point est représenté par 2 bits). Comme 2 bits ne peuvent donner que 4 couleurs, dans ce mode les points n'utilisent que 4 couleurs de la palette. Afin de pouvoir afficher 16 couleurs, on va utiliser l'astuce suivante: chaque groupe de 2 bits de l'octet va employer un des 4 groupes de couleurs: on a donc bien encore 16 couleurs, mais cette fois pas point par point, mais uniquement par groupe de points.

| | | | | |
|---|---|---|---|---|
| bits | 7-6 | 5-4 | 3-2 | 1-0 |
| couleurs | 0-3 | 4-7 | 8-12 | 13-16 |

chaque point correspond à un groupe de couleurs de la palette

### Les palettes

En mémoire, cela correspond aux octets entre $E19E00 et $E19FFF (soit 16 palettes de 16 couleurs, chaque couleur étant représentée par 2 octets). Dans ce groupe de 2 octets représentant une couleur (16 bits), on a donc 4 sous-groupes de 4 bits chacun. Chaque sous-groupe caractérise une des couleurs fondamentales, et les valeurs des bits au sein de chaque sous-groupe déterminent la nuance (le sous-groupe 3 est inutilisé):

| 1er octet | | 2ème octet | |
|---|---|---|---|
| 7654 | 3210 | 7654 | 3210 |
| vert | bleu | ---- | rouge |

Par combinaisons, on peut obtenir 4096 couleurs différentes! Les différentes nuances de vert-bleu-rouge permettent de constituer les couleurs.

recherche, supposant (à tort) que les premiers octets sont ceux de l'image. Pour vous en convaincre, essayez de visualiser des images *brushes* Deluxe Paint avec un logiciel commercial (ou avec les programmes freewares ou sharewares)... la conclusion: PicMaster GS est le **seul** à pouvoir lire et afficher correctement ces images!
D'autre part, ce programme permet de visualiser des images de taille gigantesque: en effet, le format APF permet de définir des images occupant plusieurs écrans (maximum 32768 x 32768 points). Si vous possédez suffisamment de mémoire, PicMaster GS affichera de telles images et les flèches vous permettront de faire défiler le dessin (touches 1 à 9 pour des déplacements de 1 à 9 points, touche 0 pour des déplacements page par page).
Comment produire de telles images? Par exemple avec le matériel de digitalisation Quickie (également avec Appleworks-GS, quand il ne plante pas: dites-lui par exemple que vous imprimez à l'italienne en réduction 50%, ou que vous avez une Laser et que vous imprimez à 25%). Si votre GS n'a pas assez de mémoire pour afficher l'image dans sa totalité, PicMaster GS n'en affichera que la partie supérieure.

## 3- Présentation des fichiers sources (

Pour ceux qui veulent seulement utiliser le programme, il suffit de lancer l'application PicMaster GS à partir du Finder. Pour ceux qui veulent apprendre, sur la disquette se trouve le source complet de PicMaster GS que vous pourrez étudier et modifier à loisir.
Les sources sont éditables et compilables avec TML Pascal II (ou Complete Pascal II moyennant une modification mineure: changer TMLUTILS en CTIUTILS dans les

déclarations USES de chaque source). Pour re-compiler le programme PicMaster GS, il faut les sources suivants:
- *GlobalsPicU.p*: fichier Unit contenant les variables globales
- *LoadPicU.p*: fichier Unit contenant les routines de chargement des images
- *SavePicU.p*: fichier Unit contenant les routines de sauvegarde des images
- *SaveRezU.p*: fichier Unit contenant les routines de sauvegarde des Resources $C02
- *LoadSavePic.r*: fichier ressource
- *LoadSavePic.p*: fichier principal

Méthode à suivre pour re-compiler les sources:
- compiler GlobalPicU.p
-> GlobalPic.p.o
- compiler LoadPicU.p
-> LoadPic.p.o
- compiler SavePicU.p
-> SavePic.p.o
- compiler SaveRezU.p
-> SaveRez.p.o
- à partir de LoadSavePic.p faire Add Resources dans le menu Special et indiquer LoadSavePic.r puis compiler: on obtient l'application LoadSavePic (qu'on renomme ensuite en PicMaster).

Pour ceux qui voudraient modifier les Ressources:
- soit vous utilisez l'Editeur de Ressources intégré de TML Pascal: dans ce cas seules certaines ressources sont modifiables et dans des limites assez contraignantes, mais pour un débutant, cet éditeur convient tout à fait...
- soit vous utilisez un Editeur de Ressource comme Genesys: vous pouvez modifier les Ressources de LoadSavePic.r (ou même de l'application PicMaster GS) beaucoup plus aisément qu'avec l'éditeur du TML.
- enfin pour bien comprendre la structure des Ressources et les définir à son aise, le mieux reste d'utiliser APW.REZ et d'éditer le fichier LoadSavePic.s; la compilation se faisant avec le script MAKE. A cette fin, il faut avoir les fichiers suivants:
- le script MAKE pour la compilation
- le source LoadSavePic.s
- le source Pic.Src
- la librairie Types.Rez: le préfixe d'accès à cette librairie est à indiquer dans le premier £include du fichier LoadSavePic.s.
- la librairie Types.Rez.Aux: ce sont des ajouts et corrections d'erreurs par rapport au Types.Rez original. Cette librairie est sur la disquette ToolBox-Mag.

Attention: la compilation des ressources avec APW.REZ doit se faire avec la version datée du 1-04-90. En effet, les versions antérieures distribuées aux développeurs comportaient des bugs (les ressources générées étaient parfois erronées, surtout pour celles de grande taille).


FastCopy Macintosh.

- Quel est le programme de copie de disquettes Macintosh le plus rapide du marché?

- Photonix II pour l'Apple II GS. JYB


## 4- Commentaires à propos du source (ééé)

### a- LoadSavePic.p

Le programme principal est constitué des routines d'initialisation du DeskTop, des routines de gestion des menus, de la routine d'affichage des explications et des routines d'affichage des images ressources.

#### Utilisation du Memory Manager

Dans la boucle principale, il faut noter l'utilisation d'un AuxId déterminé à partir de l'identifiant retourné par le Memory Manager: ceci est le seul moyen de pouvoir purger efficacement les Handles employés par le programme

```
gMyMemoryID:= MMStartup;            {identifiant
retourné par le MemoryManager}
auxID:= BOr(gMyMemoryID,$0100);     {identifiant
utilisateur}
......
DisposeAll(auxID);                  {purge des handles
employés avant le Quit}
```

Dans cette boucle principale, on teste également à chaque réservation de mémoire si une erreur est retournée: dans ce cas, on force le programme à s'interrompre pour cause de mémoire insuffisante.

```
SaveHdl := NewHandle($7E00, auxID, attrlocked, nil);
{réserver mémoire}
if _ToolErr<> NoError then NoMemory;     {manque de place -
> on quitte}
```

#### Procédure LoadFonts

Ensuite on initialise quelques variables globales, puis on fait appel à LoadFonts. Cette procédure a été écrite à la demande de Jean-Yves Bourdin: en effet, le fringant Rédacteur en Chef se distingue (mais il n'est pas le seul) en utilisant des fontes 'personnelles' sur son système de boot. Concrètement, il ne dispose pas des fontes Geneva et Courier standard... fâcheux, car PicMaster GS utilise ces fontes dans l'affichage du dialogue A Propos. Dans ce cas, QuickDraw emploie une fonte 'voisine': et bien entendu on a un affichage déplorable. Ayant été mis au courant de cet inconvénient, je décidai d'utiliser mon propre jeu de fontes, et pour n'avoir aucun ennui, je les stockai sous forme de Ressource.

[*NDLR: Le malheur a voulu que même avec ces changements, le résultat a été jugé un peu laid pour Toolbox-Mag, et qu'il faut bien que les navettes s'arrêtent un jour. Dans l'application PicMaster GS de la disquette Toolbox-Mag, l'alerte 'A Propos' a donc été refaite en éditant directement la ressource. En compilant les sources suivant la procédure indiquée dans l'article, vous retrouverez l'alerte originale, et pourrez comparer. De toute façon, le procédé de programmation de Bernard est intéressant, car la seule fonte qu'une application est en droit d'attendre dans le système de l'utilisateur est la Shaston 8 de la Rom, accélérée dans son affichage par FastFont. Elle doit donc charger elle-même, au besoin, toute autre fonte.*]

Pour définir les fontes sous forme de Ressources, j'ai utilisé une routine semblable à celle convertissant les images (voir le module SaveRezU.p). Ensuite, il ne restait plus qu'à écrire un module de chargement de ces Ressources et de reconnaissance par le Font Manager.

On commence par charger la Ressource, puis on ajoute la Famille

et le Numéro dans la liste des fontes du Font Manager:
*FontName:='MyGeneva';*
*FamilyNum:=$269;*
*MyFontHdl1:=LoadResource($C800, kMyFontGen10);*
*HLock(MyFontHdl1);*
*AddFamily(FamilyNum, FontName);*
Ensuite on transfère la fonte dans un Handle non purgeable, on libère la mémoire de la Ressource (ReleaseResource), puis on indique la localisation de la fonte au Font Manager:
*AddFontVar(FontHndl(MyFontHdl1),0);*
Et voilà ! on procède de même pour la fonte MyCourier et on dispose alors de deux fontes pour nos affichages.
Remarque: les fontes utilisées dans ce programme ont été légèrement modifiées (pas de nom en en-tête).
Lorsque nous afficherons le message du dialogue A Propos, le Font Manager devra procéder à des calculs de mise à l'échelle des fontes MyGeneva et My Courier (on affiche en effet ces fontes en plusieurs tailles et styles). Ces calculs ne sont effectués que lors du premier affichage et prennent un certain temps: ce qui fait que lors de la première ouverture du dialogue 'A Propos', il se passe un délai entre l'affichage du cadre et le remplissage du texte formaté. Pour éviter cela, il faut pré-charger les fontes dans leurs tailles utilisées: c'est ce que l'on fait en fin de procédure LoadMyFonts:
*InstallFont(MyFont,0);*

## Procédure LoadAResource

La procédure LoadAResource utilise la fonction SFGetFile2 afin de sélectionner le fichier ressource à explorer. Afin de ne laisser apparaître dans la liste du dialogue que les fichiers Ressources de type $B3 on utilise un TypeListPtr et une fonction de filtrage. Le TypeListPtr permet de n'autoriser que les fichiers de type $B3 (applications GS/OS) à l'affichage dans le Standard Dialog. Ces applications ne contenant pas forcément de Resource Fork, il faut le déterminer avec une fonction de filtrage testant si le fichier est de type étendu ou non.

```
gTypeList.numEntries := 1; {un seul type autorisé}
gTypeList.FileTypeEntries[1].flags :=$8000;
gTypeList.FileTypeEntries[1].FileType := $B3;     {uniquement
les applications}
gReplyRec.nameDesc := refIsNewHandle;
gReplyRec.pathdesc := refIsNewHandle;
SFGetFile2(120,40, {coordonnées d'affichage} refIsPointer;
ref(@'Quelle ressource charger?');
@FilterProc; {filtrage des fichiers étendus}
@gTypeList; {filtrage des types autorisés}
gReplyRec);
....
```

La fonction filtre étant applicable à une fonction de la ToolBox, il est impératif d'utiliser la directive {$DefProc}. Le manuel de référence indique qu'une fonction de filtrage sur un Standard Dialog retourne les valeurs suivantes:
0: aucun fichier affiché
1: affichage en mode non sélectionnable (en grisé)
2: affichage en mode sélectionnable
Voici donc la fonction de filtrage:

```
{$DefProc}
Function FilterProc( var DirParms: dirEntryRecGS): integer;
begin
   if BAnd(DirParms.flags, $8000) = IsFileExtended
   then FilterProc:=2
  else FilterProc:=1;
end;
```

La procédure de chargement des ressources s'articule ensuite sur l'exploration du fichier ressource afin de détecter celles du type $C02:
On commence par ouvrir le fichier Ressource à explorer:
*fileID:=OpenResourcefile(0,nil,@PicPath^^.bufstring);*
sachant que lors d'une recherche de Ressource, le Resource Manager débute son exploration à partir du fichier ressource dernièrement ouvert, on va limiter cette exploration à ce seul fichier, pour cela on indique 1 pour la 'profondeur' de recherche:
*FileDep:=SetResourceFileDepth(1);*
puis on cherche toutes les ressources de type $C02 (le nombre de ressources total d'un type donné est retourné par CountResource):
*for kResType:=1 to CountResource(rImageAPF) do*
*begin*
on charge la ressource voulue avec LoadResource: il faut indiquer le type (c'est $C02) et l'identifiant. Cet identifiant est retourné par GetIndResource [paramètres: le type ($C02) et le numéro (la variable de boucle)].
*PictHandle:=LoadResource(rImageAPF,*
*GetIndResource(rImageAPF,kResType));*
ensuite on décompacte les data et on transfère classiquement l'image en lieu sûr avant de revenir au menu principal (PtrToHand...). Lorsque toutes les images ont été visionnées (ou qu'on a interrompu le show en tapant ESC afin de conserver une image précise en mémoire), on referme le fichier ressource:
*CloseResourceFile(FileID);*
et bien entendu on ré-autorise la recherche sur tous les fichiers Ressources (profondeur $FFFF):
*fileDep:=setResourceFileDepth($FFFF);*

## Procédure ShowHelp

Une autre procédure mérite d'être commentée: ShowHelp qui permet d'afficher un texte d'explications dans une fenêtre avec ascenseurs.
Comme nous utilisons les Ressources, on a défini une fenêtre avec une liste de contrôles. Si on s'en tient à ce qui est indiqué dans le Window Manager, un appel NewWindow2 est suffisant... oui mais la Technote 82 Apple nous explique qu'un bug du Control Manager empêche (entre autres) NewWindow2 de fonctionner correctement si une liste de contrôles lui est associée (il y a un problème de Port). La parade consiste à définir une fenêtre sans contrôles, puis à définir indépendamment une liste référençant des contrôles et à associer cette liste ultérieurement à la fenêtre. Voici comment faire:
on récupère le port courant:
*OldPort:=GetPort;*
on ouvre la fenêtre vierge:
*Helpwind:=NewWindow2(nil,0,@DrawcontentHelp,nil,*
*RefIsResource,ref(kWindHelp),rWindParam1);*
on force le port sur la fenêtre:
*setPort(helpWind);*
on charge la liste de contrôles et on l'associe à la fenêtre:
*MyHandle:=NewControl2(helpWind,ResourceToResource,*
*ref(kCtrlListHelp));*
on dessine les contrôles avec la séquence
*BeginUpdate(helpWind);*
*DrawControls(helpWind);*
*EndUpdate(helpWind);*
puis on gère le contenu du TextEdit (routine fournie par TML dans un exemple sur la disquette accompagnant le compilateur).
Lorsqu'on a fini, on n'oublie pas de restaurer le port:
*SetPort(OldPort);*
A noter également la définition d'une procédure de rafraichissement du contenu de la fenêtre: DrawContentHelp (procédure passée comme paramètre dans NewWindow2); et comme cette procédure concerne une fonction ToolBox, on est obligé de mettre la directive {$DefProc}.

## b- GlobalPicU.p

Cette Unit contient les variables globales et deux procédures: OneError affichant le code d'erreur éventuel lors d'un appel à une fonction ToolBox; et NoMemory qui indique à l'utilisateur qu'il ne dispose pas d'espace mémoire suffisant pour utiliser PicMaster GS. Ce type de procédure devrait être utilisé systématiquement afin que chaque utilisateur sache pourquoi un programme ne fonctionne pas chez lui, plutôt que d'entendre un dzoinnnnng ou de voir son GS tourner en rond... La plupart du temps, il suffit de libérer l'espace RAM DISK ou de désactiver des NDA pour avoir à nouveau un GS capable de lancer une application nécessitant quelques milliers d'octets.
Bref, dans NoMemory, j'utilise HALT qui est le moyen de quitter l'application en cours à tout moment.

## c- LoadPicU.p

Cette Unit est chargée de charger les images, de les afficher et de permettre les changements de résolution ou de faire défiler l'image si celle-ci est supérieure à un écran. Comme nous n'employons pas le WindowManager avec des ascenseurs, c'est à nous de gérer le défilement.

### Chargement des images

On sélectionne l'image à partir d'un Standard Dialog classique. Lorsque l'image à charger a été définie, on utilise les fonctionnalités de GS/OS pour procéder à la lecture des data.
Chaque fonction GS/OS est appelée avec des paramètres: le pcount indiquant le nombre de paramètres pour la fonction, suivi des paramètres effectifs. A priori rien de particulier dans cette succession d'appels, un examen attentif du listing doit suffire à la compréhension. Cependant, il est impératif dans ce genre de travail de tester systématiquement les erreurs éventuelles de lecture. Après chaque appel GS/OS on fait donc un test de _ToolErr, on affiche le numéro d'erreur avec la procédure OneError et on quitte la routine en remettant les choses en l'état. Remettre les choses en l'état consiste à libérer les Handles réservés et à faire un Close.
Selon l'appel GS/OS les handles à libérer seront différents: on a donc la solution de faire un DiposeHandle pour chaque Handle et ce dans chaque test d'erreur (ceci est gourmand en code: voir par comparaison les procédures de sauvegarde dans l'Unit SavePicU.p) ou alors... d'utiliser GOTO. Avec GOTO on peut se brancher sur un endroit précis du programme: mais il est tout de même conseillé de ne pas abuser de ce type de programmation qui rendrait vite le source illisible par des branchements dans tous les sens. A n'employer qu'avec parcimonie donc.
On définit d'abord un label avant les déclarations de types ou variables dans la procédure puis à tout moment on met GOTO label et le branchement se fait sur l'endroit voulu. A mon avis, c'est la solution idéale de gestion des erreurs dans une procédure complexe.

### Décompactage

Lorsque l'image est chargée, on décompacte les data. Le plus simple d'abord:
*les images de type $C1* étant au format écran, un simple transfert des octets suffit.
*les images Paint*: de type $C0 avec un auxtype $0 (ou $8000 pour celles générées par GS Paint). Ces images sont constituées comme suit:
-32 octets pour la palette utilisée
- 514 octets pour la définition des motifs
- les data compactées
on va donc procéder méthodiquement:
- transfert de la palette:
*BlockMove(pictPtr,ptr($E19E00),32);*
- on avance de 546 octets (514 + 32) pour pointer sur le début d'image
*PictPtr:=ptr(ord4(pictPtr)+$222);*
- on définit les paramètres de décompactage:
*FileSize:=fileSize-$222;*
*{taille des data à décompacter}*
*UPagePtr:=ptr($E12000);*
*{pointeur sur zone de destination}*
*UpageSize:=160*200;*
*{taille de zone destination: 200 lignes de 160 pts}*
*j:=UnPackBytes(PictPtr, FileSize, UpagePtr, UpageSize);*
et voilà, le décompactage est effectué par UnPackBytes!
*les images SPF*: type $C0 auxtype $1. Ces images sont constituées comme suit: compactage de toute l'image, palette comprise. On va donc procéder comme pour le compactage PNT, mais avec des paramètres légèrement changés (UpageSize est égal à $FFFF et la taille des data à décompacter est FileSize).
*les images APF*: type $C0 auxtype $2. Ce format de sauvegarde est le plus riche et aussi le plus complexe. Fondamentalement, il est constitué comme suit:
- 4 octets: longueur du bloc MAIN contenant les images
- la chaine MAIN (précédée d'un octet de longueur)
- 2 octets: le mode de résolution
- 2 octets: le nombre de pixels par ligne
- 2 octets: le nombre de palettes utilisées
- les palettes (c'est à dire 32 octets fois le nombre de palettes)
- la 'hauteur' de l'image (le nombre de lignes)
- les descripteurs de lignes (4 octets fois le nombre de ligne): les deux premiers octets représentent la longueur de la ligne compactée correspondante
- l'image compactée ligne à ligne
- ensuite on a éventuellement le bloc PAT (les motifs): non analysé ici
- puis éventuellement aussi le bloc SCIB: non analysé ici
Ce format de sauvegarde est avantageux: on sauve le nombre de palettes que l'on veut et l'image peut avoir la taille de son choix (maxi 32768 x 32768). Cependant, comme je l'évoquais au chapitre 2 il faut faire attention: on à tout à fait le droit de mettre des informations avant la chaine MAIN; il est abusif de considérer que le format APF commence *forcément* par MAIN et la suite standard... notre premier souci est donc de rechercher la chaine MAIN et ensuite de pointer à partir de cet endroit pour analyser le compactage.
on pointe en début de zone
*UMainPtr:=Pointer(ord4(pictPtr));*
*Umain:='';*
on cherche MAIN
*repeat*
*Umain:=UmainPtr^;*
*UmainPtr:=pointer(ord4(UmainPtr)+1);*
*until Umain:='MAIN'*
ensuite on va se mettre en début de zone MAIN (avant les 4 octets de longueur)
*pictPtr:=pointer(ord4(UmainPtr)-5);*
et on va analyser chaque paramètre un à un en pointant sur les octets voulus et en déréférençant le pointeur. Par exemple, pour le 'mode':
*UmodePtr:=pointer(ord4(pictPtr)+9);*
*Umode:=UmodePtr^;*
on notera comment est déclaré un pointeur sur un type de variable particulier afin de pouvoir le déréférencer:

*UmodePtr: ^Integer; {littéralement: pointeur sur un entier}*
on procède de même pour les autres paramètres.
ensuite, vient le moment de réserver la place en mémoire pour l'image décompactée. Celle-ci va occuper plusieurs écrans: selon sa largeur (UNumPts) et sa hauteur (UNumRow).
- calcul du nombre de points: 2 ou 4 pixels par point, selon la résolution
*if UMode>= $80 then i:=4 else i:=2;*
*NumPtsApf:= UNumPts div i;*
- calcul de la largeur: nombre de fois 160 (nb de points par ligne):
*UPageLarg:=NumPtsApf div 160*
- on ajoute 1 si on a une partie d'écran:
*if (NumPtsApf mod 160) <>0 then UPageLarg:=UPageLarg+1;*
- on détermine de combien de mémoire on dispose:
*CompactMem;*
*MaxMemory:=MaxBlock;*
- on va calculer le nombre d'écrans total nécessaire pour le décompactage: nombre de fois 200 lignes (hauteur d'un écran) par nombre d'écrans en largeur
*MaxPage:=UPageLarg*(UNumRow div 200);*
- on rajoute une série si on a un nombre de lignes non multiple de 200:
*if UNumRow mod 200 <>0 then*
*MaxPage:=MaxPage+UPageLarg;*
- on va tester si cela est compatible avec la mémoire disponible (chaque page occupe $7D00 octets):
*if MaxPage>MaxMemory div $7D00 then*
*begin*
*MaxPage:=MaxMemory div $7D00; { nb écrans maxi possible }*
*InitCursor;*
*i:=AlertWindow(4,nil,Ref(4));  { alerte indiquant le manque de mémoire }*
*HideCursor;*
*end;*
- mais il faut qu'on ait un nombre entier d'écrans en largeur pour la dernière série affichable
*MaxPage:=UPageLarg*(MaxPage div UPageLarg);*
- enfin on teste qu'on peut bien afficher au minimum une série d'écrans en largeur: si ce n'est pas le cas, on quitte
- maintenant que l'on sait combien de pages réserver, on crée le handle correspondant. On a besoin de MaxPage*$7D00 octets. En Pascal, il faut d'abord effectuer l'opération avant de l'employer dans une expression. On ne peut pas écrire, par exemple:
*WriteLn($7D00*3);*
sous peine d'avoir un résultat erroné; mais on doit écrire:
*R:=$7D00*3; WriteLn(R);*
On peut également contourner la difficulté en faisant une suite d'additions. Ce type de calcul sera identique lorsqu'on déplacera le pointeur de xxxx octets: afin que les changements de bancs se fassent correctement, on additionnera xxxx fois 1 octet (ce qui évite de passer par des variables intermédiaires de calcul).
- il est temps de calculer le nombre de lignes effectivement affichables: si on ne peut afficher toute l'image (mémoire insuffisante), il faudra donc ne décompacter que le nombre de lignes autorisé:
*if UNumRow>200*(MaxPage div UPageLarg) then*
*UNumRow:=200*5MaxPage div UPagLarg);*
- étape suivante, mettre à zéro (couleur blanche) la zone réservée: en effet, si une ligne décompactée n'utilise pas toute la largeur d'écran, il faut que les points restant soient ré-initialisés. On transfère donc une page blanche dans tout le handle, par tranche de $7D00 octets.
- avant de procéder au décompactage (processus assez long), on affiche un message dans la fonte Shaston 16
- on déréférence le handle: une fois pour le transfert, une deuxième fois pour conserver la valeur de l'adresse (en vue du passage à la ligne suivante)
*UPagePtr:=UApfHdl^;*
*UApfPtr:= UApfHdl^;*
- on décompacte:
*for i:=0 to UNumRow-1 do {décompacter UNumRow lignes}*
*begin*
*UPageSize:=UNumPts;*
*USizeRow:=USizeRowPtr^; {déréférencer le pointeur de taille}*
*k:=UnPackBytes(UStartPtr, {début de la ligne compactée }*
*UsizeRow, {nb d'octets compactés }*
*UPagePtr, {début de la zone de destination }*
*UPageSize); {nombre de points de destination }*
*{ on repositionne sur le début de la prochaine ligne à décompacter }*
*UStartPtr:= pointer(longint(UStartPtr)+USizeRow);*
*{ on se positionne sur l'octet de longueur de la prochaine ligne }*
*USizeRowPtr:=pointer(ord4(USizeRowPtr)+4);*
*{ on va se positionner en début de ligne suivante: en effet, UnPackBytes positionne le pointeur en fin de zone décompactée; mais dans notre cas, on ne sait pas si cela correspond au début de la ligne suivante: on peut très bien avoir une ligne partielle horizontalement. On va donc utiliser le pointeur UApfPtr non modifié et l'incrémenter de UPageLarg fois 160 }*
*for k:=1 to UPageLarg do*
*UApfPtr:=pointer(ord4(UApfPtr)+160);*
*UPagePtr:=UApfPtr;*
*end;*
- lorsque le décompactage est terminé, on transfère le premier écran sur la page d'affichage, on initialise les variables de position et on passe à la procédure Change Mode qui permettra le défilement et le changement de résolution.

### Procédure ChangeMode

Pour cela, examinons la procédure ChangeMode:
Cette procédure est basée sur une boucle GetNextEvent: si l'événement est de type AutoKey ou KeyDown alors on analyse la touche frappée et on traite; si l'événement est de type MouseDown alors on sort de la boucle. Si la touche frappée est 1...9, les déplacement sont de 1...9 points; si on tape 0, les déplacements sont d'un écran (200 lignes ou 160 colonnes). Si on tape RETURN et que l'image occupe plus d'un écran, on obtient une copie réduite de l'image (on quitte en cliquant).
On vérifie que la ligne et la colonne sont dans les limites permises, puis on va pointer en début de zone à afficher: cette zone commence (StartLine*(UPageLarg * 160) + StartCol) octets après le début du handle. Dès lors, on transfère 200 lignes de 160 octets, en prenant soin à chaque transfert de passer à la ligne suivante (pour l'affichage, et en mémoire)!
Ouf... Ces routines de décompactage-affichage permettent de charger des images de n'importe quelle taille, quelle que soit la mémoire dont vous disposez: dans le pire des cas, une partie seulement de l'image sera affichée (NB: il aurait été possible de faire gérer la mémoire dynamiquement, on n'aurait chargé que la partie à afficher... mais cela aurait considérablement alourdi le source. Pour une plus grande clarté, j'ai préféré cette limitation).

**Entendu à l'Expo**

"Sur ce modèle de Macintosh, le clavier est en option".

## d- SavePicU.p

Cette Unit permet de sauver les images dans le format de votre choix. La partie sauvegarde proprement dite est calquée sur le chargement et ne recèle donc aucune particularité. Cependant, afin de sélectionner le type de format de sauvegarde, on emploie un Custom StandardFile Dialog. C'est à dire que le fonction SFPPutFile2 va nécessiter la définition d'un DialogHookPtr2 et d'un DlgTempPtr.
La fonction s'écrit ainsi:
*SFPPutFile2(175,25,nil,refIsPointer,ref(@SaveString), refIsPointer,ref@SaveName),DlgTempPtr(@FileTemplate)^, @DialogHookPtr2,gReplyRec);*
La procédure DialogHookPtr2 étant appliquée à une fonction ToolBox, on est obligé d'employer la directive {$DefProc}. Cette procédure sert à gérer la modification des boutons radios: on en récupère la valeur dans un tableau SaveBut[] créé à cet effet. La définition du FileTemplate obéit à une règle précise: les contrôles standards sont définis en premier et dans un ordre précis: boutons (enregistrer, ouvrir, fermer, lecteur, annuler), ascenseur, les rectangles, ligne d'édition du nom de sauvegarde, taille disponible, bouton 'Dossier' et ensuite seulement les ajouts de votre cru. Etudiez attentivement le listing pour comprendre le fonctionnement de la procédure. A noter que lorsque les Ressources n'étaient pas encore implémentées (système GS/OS 4.0), il était obligatoire d'utiliser une telle procédure pour définir des boutons en français.
Dans les procédures de sauvegarde, vous noterez comment sont gérées les erreurs: pas de GOTO Label cette fois, mais un code plus important.

## e- SaveRezU.p

Cette Unit permet de définir une Ressource Image et de la sauver soit en Source compilable, soit directement dans un fichier Ressource. On notera que la compilation d'une ressource volumineuse par APW.REZ génère des erreurs (octets non conformes).
Le Custom StandardFile Dialog est construit sur le même modèle que dans l'Unit SavePicU.p (mais on emploie des CheckBox au lieu de Radio Button).
Lorsque le nom du fichier de sauvegarde et son/ses types de sauvegarde sont définis, le programme permet de découper le rectangle englobant la zone écran à sauver sous forme de Ressource $C02. Le rectangle est défini par la procédure DoClip. Une fois le rectangle déterminé, on ajuste les coordonnées de manière à avoir un nombre de pixels multiple de 4; puis on compacte le rectangle avec PackBytes.
La procédure DoClip est assez curieuse: elle permet de découper un dessin en mode 320 alors que l'Event Manager est initialisé en 640. Ceci ne serait pas possible si on utilisait le Window Manager afin d'afficher les images: il faudrait ré-initialiser le système en mode 320 ou 640 lors des changements de résolution (nous verrons cela dans un prochain article).
La procédure utilise GetNextEvent afin de tester les évènements: un MouseDown permet de quitter la boucle et on récupère les coordonnées de la souris grâce à TaskMaster.

### Sauvegarde de la ressource compilée

On tente d'ouvrir le fichier Ressource avec le nom de sauvegarde employé; si une erreur survient, c'est que le fichier n'existe pas: dans ce cas on le crée (CreateResourceFile); sinon on ajoute notre ressource (AddResource) et on la prépare à être sauvée sur disque par UpdateResourceFile (ainsi lorsqu'on fera CloseResourceFile, les Ressources modifiées seront automatiquement sauvées sur disque). Remarquez comment on détermine le numéro d'identifiant de la ressource: on utilise UniqueResourceID afin que le Resource Manager nous fournisse un numéro non employé.
Pour la sauvegarde sous format Source, on va d'abord ajouter le suffixe .SRC au nom de sauvegarde. Lors de l'appel à DestroyGS, on teste l'erreur retournée: en effet si on a une erreur $46 (le fichier est inexistant) alors on peut le créer et sauver les data; dans le cas contraire, c'est que le fichier existe et alors on quitte (pour éviter de détruire un fichier de même nom). On pourrait faire cela de manière plus 'propre' en demandant un nom de sauvegarde du fichier Source à l'aide d'un StandardFile Dialog (ceci peut constituer un exercice -simple- pour le lecteur).
La sauvegarde de la ressource en format source va se faire comme suit:
- ouverture d'un fichier REZ (type $B0 auxtype $15)
- écrire '£Define rImageAPF $C02' pour que le compilateur APW.REZ puisse ultérieurement employer la variable rImageAPF.
- écrire 'data rImageAPF xxx' (avec xxx numéro d'identifiant)
- puis on convertit les octets de la Ressource en leur équivalent Hexa et on sauve le tout par paquet de 32 octets (avec $ en début et la chaîne entre guillemets).
L'étude du listing permettra de comprendre comment on procède.

## f- LoadSavePic.s

C'est le source du fichier Ressource employé lors de la compilation. Ce source est écrit avec l'éditeur APW et compilable avec APW.REZ. Ce source est chaîné avec Pic.Src (contenant l'image compactée d'introduction et les fontes MyGeneva et MyCourier).
A noter: pour que la compilation soit possible, il faut indiquer en début de source dans une directive £Include le préfixe d'accès complet à la librairie Types.Rez (pour la compilation de notre source, utilisez en plus la librairie TypesRez.Aux fournie sur la disquette ToolBox-Mag, car elle est mise à jour par rapport à celle fournie par Apple).
La suite du listing est simplement le source des Ressources employées par notre application.
Il est également possible de modifier les Ressources avec Genesys ou l'éditeur TML en éditant LoadSavePic.R (fichier ressource généré par APW.REZ).
Toutes remarques ou suggestions seront les bienvenues: envoyez-les à la revue qui transmettra... Dans un prochain numéro, je traiterai l'affichage via les structures LocInfo (et donc à l'aide du WindowManager) ainsi que l'emploi des SCBs en FillMode afin de compléter le tour de la question.

### Entendu à l'Expo

"Je peux parler de son Font/DA Mover et de son TransProg parce qu'il les a mis en shareware. Je ne peux pas parler de son Tarot, même s'il me l'a remis personnellement, parce qu'il est édité par Toolbox.

— Comment ça, pas libre? C'est faux, je suis parfaitement libre d'écrire ce que je veux."

# Fontes GS: suite (*)

J.Y. Bourdin

## Remords

Deux erreurs dans mon article sur les fontes de Toolbox-Mag N° 1. Ce sont les risques du couper-coller entre morceaux de textes écrits à des époques différentes...

① J'écrivais que la seule machine qui égale le GS à l'impression sur imprimante à 9 aiguilles est le Mac. C'est faux: **les impressions du Mac sur ImageWriter sont considérablement inférieures à celles du GS.** Même la qualité dite "Supérieure" du driver ImageWriter du Mac est inacceptable.
Et cela pour une raison simple: le driver ImageWriter du Mac est plus rapide que celui du GS, parce qu'il est bi-directionnel (il imprime de gauche à droite et de droite à gauche, à l'aller comme au retour). Du coup, l'alignement à l'impression n'est pas bon, et la qualité globale est mauvaise.

② Si le driver GS est meilleur, c'est parce que lui est précisément unidirectionnel (il n'imprime que de gauche à droite, à l'aller seulement). C'est pourquoi j'avais bien tort (merci, Yvan) de demander un driver unidirectionnel: précisément, *nous l'avons*! Ce dont nous avons besoin en fait, c'est de pouvoir *choisir entre bi- et uni-directionnel.*
C'est justement ce que nous apporte le driver en freeware de Bill Heinemann. Si votre GS est connecté directement sur votre Imagewriter (sans passer par Appletalk) nous conseillons ce driver. Il est nettement plus rapide que le driver Apple ou que celui de Claris.

*Dernière minute: on m'a dit du bien du driver du système 5.03. Rendez-vous dans Toolbox-Mag 3.*

## Compléments

❑ Voyez la rubrique "Bidouilles GS" à propos d'un petit secret d'AppleWorks-GS pour doubler automatiquement la taille des images pour imprimer en condensé.

❑ Si vous avez une ImageWriter LQ, préparez vos "triplettes" de fontes type 12/24/48. Le driver de LQ en préparation chez Apple utilisera des fontes de taille 4 fois supérieure pour gérer la résolution de la LQ.

❑ La pratique qui consiste à avoir des fontes "pré-stylées" en gras et italiques pour avoir une bonne impression sur l'imprimante, plutôt que de laisser faire QuickDraw, n'est pas une invention rocambolesque de JYB. C'est la pratique... d'Adobe lui-même: dans la Laser, les fontes Adobe sont précisément pré-stylées en gras ou oblique, même si le dessin des fontes-écran correspondantes est généralement confié à QuickDraw.
*Il n'y a pas de secret: de toute façon, si on veut une impression de qualité, il faut faire des fontes spéciales pour l'imprimante*, qu'on passe par QuickDraw ou par PostScript.

## Regrets

*Nous avions prévu de mettre dans ce numéro d'autres "vraies fontes" Imagewriter, ainsi que deux modèles de jaquette de cassette audio AppleWorks-GS. La place nous manquant cruellement sur le disque, ce sera pour un prochain numéro. Nos excuses.*

### Unidirectionnel?

Durant AppleExpo, de nombreux interlocuteurs m'ont demandé d'ajouter le mode unidirectionnel au driver Imagewriter. Bizarre, bizarre, car le driver fourni par Apple est déjà unidirectionnel. En fait il semble que mes interlocuteurs utilisent, allez savoir pourquoi, le driver fourni par Claris. Ce dernier est une vieillerie non conforme aux spécifications 5.02.
Bon prince, je me suis remis à l'ouvrage et désormais le driver Claris peut imprimer en unidirectionnel ou en bidirectionnel, tout comme le driver Apple modifié par mes soins. En outre, sur le driver Apple modifié, si lancer l'impression en pressant PommeOuverte active le mode bidirectionnel, presser Option active le mode 'foncé' dont certains regrettaient l'absence. Il n'est pas possible d'imprimer en foncé unidirectionnel, mais est-ce bien gênant?
Ces drivers seront, comme les versions précédentes, disponibles sur GSinfos, le disque du GS Club. Yvan Kœnig

# *Supplique aux utilisateurs de GS* (*)

J.Y. Bourdin

Je l'ai déjà écrit, et ce n'est pas une plaisanterie. Le principal frein au développement de l'Apple II en France, ce n'est pas Apple France: *ce sont les utilisateurs de l'Apple II.*

Aux débuts du GS, il s'agissait de la lenteur et de l'inertie des possesseurs de 8 bits pour passer au GS. Aujourd'hui, il s'agit de la *taille de la mémoire des GS en service.*

Pourquoi Taïto, qui fait toujours des jeux magnifiques entièrement au standard Prodos (qui se lancent et se quittent, s'installent sur le dur, etc), a-t-il dû faire, avec l'extraordinaire Rastan, un logiciel en P8 qu'on ne peut pas quitter? Parce qu'un jeu comme Rastan, avec ses animations et ses sons, doit demander au moins 800k de mémoire. On ne peut pas avoir à la fois le système et Rastan dans 1,2 Mégas de mémoire. Pour mettre à la fois le système et Bouncing Bluster II dans cet espace, les auteurs ont dû faire des contorsions affolantes.
Et la conséquence est là: tant que la majorité des utilisateurs n'aura qu'un méga de mémoire, il n'est pas question de rajouter la moindre fonction à Bouncing Bluster ou à Rastan, et à bien d'autres encore.

C'est vrai, Activision ne supporte plus PaintWorks Gold. Mais la faute à qui? Le logiciel ne tourne déjà pas sous 5.0 avec 1,2 Mégas, et les utilisateurs s'en prennent à Activision au lieu de se mettre à niveau! Comment voulez-vous qu'Activision puisse penser qu'il existe un quelconque marché pour un PaintWorks Gold plus puissant, donc plus gourmand en mémoire? Et du coup, pour le scrolling horizontal dans les images, dont j'ai besoin, eh bien je peux toujours attendre. Et quand j'en aurai assez d'attendre, quelle autre solution aurai-je que de changer de machine?

Le programmeur de l'excellent programme de mathématiques appelé "GS Numerics", dont Toolbox-Mag vous rendra compte un jour, est déjà considérablement à l'étroit sur GS 1,2 Mégas. Il prépare une version Mac: cette version aura un bon nombre de fonctions en plus qui ne sont pas dans la version GS. Pourquoi? Parce qu'il y a un réel marché pour des applications de 2 Mégas sur Mac, mais pas sur GS. Conclusion: un magazine honnête comme Toolbox-Mag se doit de dire à ses lecteurs matheux: "attendez la version Mac, elle est meilleure". *Est-ce vraiment cela que vous voulez, amis?*

Utilisateurs qui avez 1,2 Mégas de mémoire, libre à vous de ramer comme un forçat sur votre GS, c'est votre affaire. Seulement, du même coup, vous me privez des développements et des logiciels auxquels j'aspire. Et là, je vous en veux: **vous bridez mon GS.**

A l'Expo, le méga de mémoire coûtait 300F. Une carte mémoire zéro K coûte moins de 800 F. Vous pouvez revendre en échange votre second lecteur (un /RAM5 de 800k fera son travail), et autres bricoles superflues, et être gagnants dans l'histoire: ce n'est donc pas une question d'argent, c'est juste de l'inertie de votre part.
Ce serait vraiment le diable si vous ne trouviez pas un commerçant capable de vous vendre à un prix correct une carte d'extension-mémoire extensible au moins à 4 Mégas et munie de 2 Mégas (éventuellement avec une reprise de votre carte Apple 1 Méga - si vous avez des puces correctes sur cette carte!).

Le GS est une machine magnifique, parce que c'est *un Apple II d'aujourd'hui, de l'époque du 16 et du 32 bits, de l'époque du méga à 300F*. S'il vous plait, cessez de l'étouffer, laissez-le montrer ce qu'il sait faire.
Mettez-vous à jour: vous n'en mourrez pas, que diable! En revanche, le GS, lui, souffre déjà de votre inertie.

Quoi qu'il en soit, ne vous faîtes aucune illusion. Sur le Mac, le système 7 va contraindre tout le monde, bon gré mal gré, à étendre sa mémoire. Pour le GS, la même chose ne saurait tarder. Quand tous les traînards vont demander, contraints et forcés, de la mémoire pour leur Apple, les prix ne seront plus les mêmes. Et ***ils n'auront même pas le droit de dire qu'ils n'étaient pas prévenus!***

# Exotica

par François Hermellin (🍎)

*[NDLR: "**Exotique**", c'est **ce qui se passe ailleurs**, là-bas. Mais c'est aussi **ce dont nous rêvons** - pulpeuses métisses de Bahia ou sombre cube californien qui pourrait bien être "Our Next Computer". C'est enfin **ce qui se passe dehors**, tout simplement (et d'abord chez notre cousin germain Macintosh, mais pas seulement).*

*Quand on a un GS, c'est qu'on n'aime pas les œillères, c'est qu'on est un petit curieux qui veut tout voir et tout savoir: François Hermellin est comme ça, et nous aussi. Merci à lui d'avoir accepté de nous écrire ces "**Exotica**".]*

## Motorola en manque

Motorola a de gros problèmes avec la production de son nouveau microprocesseur, le 68040. Initialement prévue pour Juin 1990, sa disponibilité a été reportée à Décembre. Eviter le genre de problèmes qu'a connu Intel avec son 80486 serait la raison essentielle de ce retard (voulant commercialiser le plus rapidement possible, Intel avait lancé sur le marché des processeurs buggés). Le 68040 est particulièrement intéressant, car c'est le premier des chips de Motorola qui comporte des instructions RISC en plus des classiques CISC. Ce processeur dégage une puissance de 20 Mips !

## Un nouveau Cube

Le 18 Septembre dernier, Next, la firme de Steve Jobs, a présenté une nouvelle version de son Cube. Celui-ci fonctionne désormais en couleurs et est basé sur un 68040. Il comprend une puce de compression d'images dite C Cube. Du fait du retard de livraison des 68040, Next ne devrait pas pouvoir livrer ses nouvelles machines avant le début de l'année prochaine.

## Pomme contre pomme

Apple a, il y a de cela quelques années, passé un accord avec la société qui défend les droits des Beatles et qui porte également le nom d'Apple. Apple, pour pouvoir continuer à utiliser son nom, a dû verser une somme d'argent et surtout promettre de ne pas s'attaquer au domaine musical.

Avec la sortie des nouveaux Macs d'entrée de gamme et leur micro intégré, Apple aurait limité le temps de digitalisation des sons à 10 secondes pour éviter tout problème juridique!

## Pomme... Pomme... Pomme

Pour rester dans le domaine du son, il semblerait qu'Apple travaille actuellement sur un système appelé *Voice Commander*. De même que la souris a permis d'améliorer la communication homme-machine qui ne passait jusque là que par le clavier, le microphone et la reconnaissance de la parole œuvreraient dans cette voie. Les ordres reconnus par le Mac resteraient assez simples ("fermeture" de la fenêtre au premier plan par exemple).

## La France, 2ème marché mondial pour les produits Mac

L'importance de notre marché pousse de plus en plus les acteurs principaux du marché américain à s'intéresser à nous. D'une part, les distributeurs US, avec leur forme de vente massive par correspondance, débarqueraient en France de façon beaucoup plus importante en 1991. Apple étudierait la reconnaissance de nouvelles formes de distribution...

D'autre part, les éditeurs américains semblent suivre le mouvement et être de plus en plus nombreux à installer leur filiale européenne en France. Des changements de stratégies vis à vis des distributeurs traditionnels vont avoir lieu, tels celui de Claris par rapport à P-Ingénierie.

## Le Mac, nombril du monde ?

Lors de la remise des trophées Apple (AppleExpo), les fanas du Mac qui s'auto-congratulent un peu trop facilement, ont découvert, émerveillés, avec *Direct Acces de Martin Scheffer* ce que nous connaissons depuis longtemps avec... *AppleWorks-GS*! L'auteur du très bon File Guard a conçu un système similaire qui permet de faire passer des infos d'une fenêtre à l'autre, sans avoir à transiter par le presse-papiers. Le logiciel, distribué par Alcyd, rajoute cependant d'autres options intéressantes. En phase de débuggage, il ne devrait pas être commercialisé avant un petit bout de temps.

## Adobe s'attaque au Fax

Après avoir mis au point un langage de description de pages (PostScript I puis II, qui sait désormais gérer la couleur, l'alimentation des bac-papiers, le recto-verso ...), de description d'écran (Display PostScript, licencié notamment à IBM et à Next), Adobe travaille actuellement avec des constructeurs nippons sur une interface PostScript permettant de gérer des télécopieurs Groupe III. Les premiers devraient voir le jour début 91 chez NEC et permettre d'avoir une impression par télécopie de qualité.

## Atari joue dans la cour des grands

Cette firme vient de sortir une station de travail proche des Mac II CI et FX. L'Atari TT est basé sur un 68030 à 32 Mhz avec 8 Mo de RAM, extensible à 26 Mo, 512 Ko de ROM, 2 x 256 Ko de RAM-cache et 48 Mo de disque dur. Si ces capacités semblent intéressantes a priori par rapport à son prix (22 000 F HT), le système d'exploitation (GEM et ses bugs !) et la bibliothèque de logiciels (celle du ST) le sont moins. Unix System V (version 4) ne sera disponible que "courant 91". Et l'on se demande qui commercialisera cette machine, vu les piètres connaissances, dans l'ensemble, des revendeurs Atari plus habitués à vendre des "Space Invaders" avec leurs ST que des stations de CAO.

## Les lacunes de Windows 3.0

Les lacunes de cette interface graphique pour "brouette de chantier" sont tellement nombreuses que des cohortes de programmeurs ne cessent de produire de nouveaux utilitaires pour pallier ses défauts. Les derniers s'appellent StraightLine, OnTop, Firstapps ...

## Des softs moins chers

Avec l'apparition de nouveaux Macs bas de gamme, la stratégie des éditeurs a dû évoluer. La plupart d'entre eux proposent soit des versions légèrement bridées mais nettement moins chères (par exemple, les produits Light de WinSoft), soit des prix très attrayants pour les étudiants. En effet, une personne qui acquiert une machine à 6 500 F n'acceptera jamais d'acheter un logiciel qui coûterait aussi cher que sa machine. D'après certaines études, le prix d'un logiciel ne devrait pas dépasser 30 % du prix de l'ordinateur. La baisse de prix du hard a entraîné celle du soft. De même, ce mouvement devrait toucher les périphériques.

## Mac: la saga des systèmes

Après la sortie du système 6.06 qui permet de gérer notamment l'entrée micro des nouvelles machines Apple, on attend toujours le système 7.0. Celui-ci devrait voir le jour en début d'année prochaine. Entre temps, des systèmes 6.07 et 6.08 devraient corriger les multiples bugs du 6.06 actuel. Pour en revenir au 7.0, il semble que le nombre et la complexité des divers périphériques que l'on peut connecter à un Mac aient rendu les opérations de tests et de débuggage plus longues que prévues.

Le système 8.0 serait, quant à lui, bien avancé. Mais il poserait un problème à la division marketing d'Apple. Système d'exploitation de la prochaine génération d'ordinateurs Apple, il tournerait sur des machines basées sur des processeurs RISC et/ou à architecture parallèle.

En contrepartie des performances obtenues, la compatibilité avec les Mac/OS actuels serait compromise en partie ou totalement.

Question à 50 francs : Comment faire avaler aux millions d'utilisateurs de Mac la pilule amère de la perte de la compatibilité avec les anciens systèmes d'exploitation, quand on n'a déjà pas réussi à la faire avaler aux utilisateurs de GS?

## Entendu à l'Expo

"Le prix public de cet article est de 2360 Francs TTC. A l'occasion de l'Apple-Expo, nous faisons un tarif spécial, qui est de...

Attendez, je cherche ma feuille de tarifs... Marcel, tu as pris les tarifs expo? Oui, voilà: 2380 Francs TTC"

(le client paye 2380 F avec le sourire).

# GS-Puzzle : un puzzle en TML Pascal

par Bruno Boissière ( )

*[NDLR: Mr Boissière, vous vous en souvenez peut-être, était l'auteur d'un très bon puzzle écrit en GS Basic et publié dans le regretté Tremplin-Micro. A l'occasion de l'Apple-Expo, il nous en a présenté la version en TML Pascal II. Nous en avons profité pour lui demander de faire profiter les lecteurs de Toolbox-Mag de son expérience de la migration vers le TML.*
*Vous trouverez donc trois parties dans son article: la première est tout simplement la documentation du programme, pour ceux qui veulent jouer au puzzle sans se soucier de programmation. La seconde est une comparaison entre GS Basic et TML Pascal II, qui tourne à l'avantage du second. La troisième est un ensemble de conseils et d'avertissements pour le programmeur en TML Pascal II.*
*Reste bien entendu une quatrième partie: le source de l'application. Vous trouverez le source complet en deux versions différentes: l'une, sur la disquette "normale" de ce numéro, appelée "Puzzle.p", est le source avec les accents. Attention, ce source est compilable directement depuis TML Pascal II, mais celui-ci ne saura pas l'imprimer correctement, à cause des accents.*
*L'autre, appelée "PuzzleSa.p", est un listing sans accents pour une sortie sur imprimante correcte depuis l'éditeur de TML Pascal. Vous trouverez ce second source sur la disquette "bis" de ce numéro, ainsi qu'une version en fichier texte Ascii GS, si vous voulez l'imprimer avec des fontes graphiques ("Puzzle1.Text"). Le source est complété par un fichier ressources éditable sous TML, "Puzzle1.R".]*

## ① Comment jouer à GS-Puzzle ( )

Ce jeu est un puzzle informatique où les pièces sont découpées dans des images GS. Au moins un avantage évident: on ne sème plus les pièces n'importe où.
Après avoir lancé l'application, appelée "GS.Puzzle" sur le disque, il vous faut tout d'abord charger une image grâce à l'article "Ouvrir" du menu "Fichier". Vous en trouverez une sur la disquette, mais vous pouvez choisir toute image 320 GS.

*[NDLR: Ne maudissez pas Mr Boissière si vous trouvez cette image trop difficile. La rédaction de Toolbox-Mag porte l'entière responsabilité de ce choix sadique. Nous nous engageons à publier le nom du lecteur qui aura fait le meilleur temps, en 100 pièces, sur ce Van Gogh!]*

GS-Puzzle ne vous proposera que les fichiers de type $C1 (images non compactées) et les fichiers de type $C0 (images compactées). Pour ce dernier type de fichier, le programme ne sait charger que les fichiers de type auxiliaire $0000 (format Paint) et non pas ceux de type auxiliaire $0002 (format Apple). Il vous le signalera si vous tentez de le faire. Mais on m'a dit que vous trouveriez de quoi faire les conversions dans ce numéro même.

Attention: *GS-Puzzle fait ses affichages de texte en utilisant la palette de l'image que vous avez chargée.* Si jamais la barre de menus devient illisible, quittez GS-Puzzle par Pomme-Q, et passez dans PaintWorks Gold. Chargez l'image, utilisez la commande "Sort Palette" pour ranger les couleurs de la palette (la plus foncée à gauche, la plus claire à droite), sauvez l'image et repassez dans GS-Puzzle.

Une fois votre image chargée, vous pouvez alors choisir le nombre de pièces de votre puzzle avec l'article "Nb de pièces" du menu "Jouer". Une boîte de dialogue vous propose 4,16,25,64, ou 100 pièces. Le choix par défaut est de 25 pièces. Tant que vous n'avez pas mélangé votre puzzle, vous pouvez changer le nombre de pièces. Lorsque vous êtes prêt à jouer, mélangez grâce à l'article "Mélanger" du menu "Jouer". Laissez le temps au GS de mélanger les pièces.
Le mélange étant terminé, vous pouvez commencer à reconstituer l'image. Le déplacement des pièces se fait par échange de deux d'entre elles. Pour cela il suffit de "cliquer" sur les pièces que vous voulez échanger. Chaque fois que vous "cliquez" sur une pièce, un bip retentit et la pièce s'entoure en noir. Si auparavant vous avez cliqué sur une autre pièce, elle s'échange avec la première.
Dès que le mélange a été effectué, vous pouvez vérifier quand vous le souhaitez si vous avez réussi à reconstituer l'image. Cette vérification est accessible par l'article "Vérifier" du menu "Jouer". Si vous avez réussi, GS-Puzzle vous dira combien de temps vous avez mis.

A tout moment, pour vous aider, vous pouvez, par l'article "Image entière" du menu "Ecran", passer de l'image du puzzle à l'image entière et vice-versa avec l'article "Image Puzzle".
Vous pouvez aussi avec l'article "Grille" du même

menu faire apparaître ou disparaître une grille blanche séparant les pièces: cela peut aider à distinguer les pièces.
Dans n'importe quelle phase du jeu, vous pouvez changer d'image en en chargeant une autre ou, bien sûr, sortir du programme par l'article "Quitter" du menu "Fichier".
Faites attention aux images comportant, une fois découpées, des pièces semblables. Vous penserez les avoir reconstituées, alors que pour le programme, les pièces ne seront pas à leurs emplacements de départ.

***Amusez-vous bien !***

**TML et ExpressLoad.**

Amis programmeurs en TML Pascal II, sachez que le compilateur de TML Pascal n'est pas entièrement à jour avec GS/OS: si TML a des ressources, il ignore ExpressLoad. Si j'en juge par tous les envois de programme TML que j'ai reçus pour Toolbox-Mag, vous aussi vous êtes aussi fâchés avec ExpressLoad (ou avec APW, plutôt).

Quoi qu'il en soit, quand vous avez compilé la dernière version de votre programme, votre travail n'est pas encore fini. Si votre programme s'appelle "Programme.1", il vous reste à passer sous APW/Orca 1.1, et à taper successivement:

*COMPACT PROGRAMME.1 -O PROGRAMME.2 -P -S*
*EXPRESS PROGRAMME.2*
*DUPLICATE -R PROGRAMME.1 PROGRAMME.2*
*DELETE PROGRAMME.1*
*RENAME PROGRAMME.2 PROGRAMME.1*

Il n'existe aucune raison valable de laisser ce travail à un Rédacteur en Chef!

*JYB*

## ② De GS Basic à TML Pascal II (🍎🍎)

Lorsque je possédais un Apple II Plus, je programmais en Basic AppleSoft. Ayant acquis un II GS, afin de pouvoir utiliser la Toolbox pour réaliser des programmes possédant une interface utilisateur évoluée (menus déroulants, multifenêtrage, etc...), je m'étais mis au GS Basic.
Avec ce langage, j'ai eu l'impression de ramer un peu tout seul dans mon coin. En effet, apparemment très peu de personnes utilisent ce langage. De plus Apple a arrêté son développement. La dernière version commercialisée était toujours une version bèta. Celle-ci tourne sous GS/OS 5.0 mais on ne peut pas imprimer de listing. Il faut pour cela reprendre le système 3.1. Enfin GS Basic n'étant pas compilé, une application écrite dans ce langage ne tourne que pour l'utilisateur qui possède ce langage. Toutes ces raisons, et tout le bien que j'entendais dire de TML Pascal II, m'ont fait craquer.

En plus de sa capacité à créer des applications entièrement autonomes, TML Pascal II possède un certain nombre d'avantages sur GS Basic:
- TML Pascal II possède un éditeur très puissant. Il est pleine page et toutes les fonctions de Line Edit y sont actives. En particulier le couper-coller et le copier-coller apportent un confort d'utilisation très important. Par exemple la concaténation de deux programmes, qui était une vraie galère sous GS Basic, est ici immédiate.
- TML Pascal II étant compilé, il tourne plus vite que GS Basic. A titre d'exemple le programme Puzzle écrit en TML Pascal II met 32 secondes pour mélanger 64 pièces alors que la version écrite en GS Basic met 50 secondes.
- TML Pascal II possède un éditeur de ressources. Les lignes interminables de data décrivant les menus, les fenêtres ainsi que l'ouverture et la fermeture, si fastidieuses, des outils deviennent inutiles. Tout est décrit avec l'éditeur de ressources, et lié au programme par très peu d'instructions.
Prenons l'exemple de l'ouverture des outils.
En GS Basic, on écrit:

```
GRAF INIT 320
myid%=EXFN_MMStartUp
aa%=EXFN_loword(TEN("C005"))
hà=EXFN_NewHandle(4*256,myid%,aa%,0)
aaà=PEEK (hà)+256*PEEK(hà+1)+256*256* PEEK
(hà+2)
_WindStartUp(myid%)
_CtlStartUp(myid%,aaà)
_LEStartUp(myid%,aaà+1*256)
_DialogStartUp(myid%)
_MenuStartUp(myid%,aaà+2*256)
_ScrapStartUp
_SFStartUp(myid%,aaà+3*256)
_DeskStartUp
```

Avec TML Pascal II, l'éditeur nous permet grâce à la ressource de type "Tool startup" de cocher les outils dont on a besoin. Il suffit d'écrire les deux lignes suivantes en tête de programme:

```
gMyMemoryID:=MMStartUp
gStartStopRef:=StartUpTools(gMyMemoryID,RefIsResourc
e,Ref(1));
```

en ayant déclaré bien sûr les variables:

```
gMyMemoryID:Integer;
gStartStopRef:Ref;
```

La fermeture des outils se fait aussi facilement:

```
ShutDownTools(RefIsHandle,gStartStopRef);
```

De plus TML Pascal II est, pour un langage compilé, d'une utilisation très simple. Finis les sauts entre l'éditeur, le l'éditeur de liens et le compilateur, comme en Pascal UCSD. Ici, on ne travaille que sous éditeur. Si, à la compilation, une erreur est détectée, TML Pascal II positionne le curseur à l'endroit de l'erreur. Il n'y a plus qu'à la corriger et à relancer la compilation: génial !
J'ai quand même quelques reproches à faire à TML Pascal II:
- Il lui manque une interface évoluée avec GS/OS. Par exemple, pour charger une image avec GS Basic, on écrit:

```
OPEN aa$, FILTYP=193 AS£11, 32767
GET£11;pic!(0)
```

Ceci a pour effet de charger le contenu du fichier de nom aa$ dans la variable pic!.
En TML Pascal II, on se sert de l'Unit GSOS, et on doit

écrire:

```
parmsOpen.pcount:=2;
parmsOpen.pathname:=@FileName;
OpenGS(parmsOpen);
parmsRead.pcount:= 4;
parmsRead.refnum:= parmsOpen.refnum;
parmsRead.databuffer:=gPuzzleHndl^;
parmsRead.requestCount :=$8000;
ReadGS(parmsRead);
parmsClose.pcount:=1;
parmsClose.refnum:=parmsOpen.refnum;
CloseGS(parmsClose);
```

sans oublier de déclarer les variables:

```
parmsOpen: OpenRecGS;
parmsRead: IORecGS;
parmsClose: RefNumRecGS;
```

Ceci a pour effet de charger le contenu du fichier de nom FileName dans la variable dont le handle est gPuzzleHndl.
Peut mieux faire.

- L'inconvénient avec TML Pascal II, c'est que c'est un Pascal et donc, toutes les variables utilisées dans le programme doivent être déclarées. Le GS Basic étant un basic, il est beaucoup plus souple de ce point de vue-là. En revanche, au niveau des types de variables, quelle richesse avec TML Pascal II !
- En GS Basic, les instructions TIMER ON, TIMER OFF et SECONDSà étaient très pratiques pour réaliser un chronomètre. En TML Pascal II, il faut passer par la procédure ReadAsciiTime. La fonction qui suit donne le nombre de secondes écoulées depuis le début de la journée:

```
{Lecture de l'heure en secondes}
function Chronometre:LongInt;
var
i:Integer;
timeStr :packed array[0..19] of Byte;
chrono :Str255;
begin
ReadAsciiTime(@timeStr);chrono:='';
for i:=0 to 19 do
chrono:=Concat(chrono,Chr(BAnd(timeStr[i],$7F)));
Chronometre:=3600*String2LongInt(Copy(chrono,10,2)+
60*String2LongInt(Copy(chrono,13,2))+String2LongInt(
Copy(chrono,16,2));
end;
```

Malgré ces petits défauts, il n'y a pas à hésiter, TML Pascal II l'emporte. De plus, une nouvelle version (Complete Pascal) est disponible. Peut-être de bonnes surprises!
Pour vous mettre à TML Pascal II, deux choses sont indispensables: le livre de Jean-Pierre Curcio "Boîte à outils de l'Apple IIgs" pour maîtriser la Toolbox, et la disquette "TML Pascal Source Code Library II" qui est bourrée d'exemples de programmes.

## ③ Conseils aux utilisateurs de TML Pascal II (⌘⌘).

Dans cette partie, je tâcherai d'exposer quelques problèmes que m'a fait rencontrer la programmation en TML Pascal II:
- L'éditeur de ressources est bogué. Lors de modifications, il peut créer quelques fois des fichiers ressources qu'on ne peut plus relire. Comme, sous l'éditeur de ressources de TML Pascal II, il est impossible de sauver le fichier sous un autre nom ou de l'imprimer, on a intérêt à faire des duplications régulièrement.
- Quand le source devient un peu long, en supprimer un morceau devient une vraie galère. En effet une fois sur deux, le "delete" du morceau perturbe la suite du programme. Ce problème est signalé dans la documentation de TML Pascal II (feuille volante).

Au lieu de prendre des risques, je sauve donc mon source sous un nom bidon puis j'ouvre un nouveau source avec le nom du premier. Il ne me reste plus qu'à faire des copier-coller des parties que je veux garder. C'est plus long qu'un "delete", mais c'est parfaitement fiable et sans risque.

- L'interface de TML Pascal II avec la Toolbox redéfinit les arguments de certaines instructions. Voici quelques exemples:

• Avec TML Pascal II, les paramètres à passer à PaintPixels définissent la destination par un pointeur sur un point (PaintParam.ptrToDestPoint) et non pas sur un rectangle comme dans la Toolbox.
• D'après TML Pascal II, le troisième argument de GetIText et SetIText doit être un Str255, alors que d'après la Toolbox cela doit être un pointeur.
• UnpackBytes dans TML Pascal II attend un pointeur en troisième argument, alors que pour la Toolbox c'est un handle.

Il y en a sûrement bien d'autres. Attention donc à la nature des arguments !

- J'ai rencontré des problèmes dans le calcul d'expressions arithmétiques mêlant des entiers avec des réels. Les calculs donnaient n'importe quoi. J'ai résolu le problème en transformant, avant calcul, les entiers en réels par des changements de variables (varReel:=varEntier). Ainsi les calculs ne se font que sur des réels et tout va bien.
- Avec 1,25Mo de mémoire, la compilation en mémoire de sources importants (plus de 40Ko) pose problème. Soit on revient à l'éditeur sans que le programme ait été éxécuté, soit on a droit à un "Out of memory". Dans ces cas là, il faut compiler sur disque ou avoir plus de 1,25Mo de mémoire.
- L'impression de sources contenant des lettres accentuées pose des problèmes. Non seulement elles ne sont pas imprimées, mais les 'é' font passer l'imprimante en mode dilaté.
- Avec la version française de GS/OS 5.0 et l'option traduction du tableau de bord sur Français: pas de problème pour les lettres accentuées, ni pour les parenthèses, ni pour les crochets. En revanche, l'arobas (@) est inaccessible. Pour l'avoir il faut mettre traduction sur standard. Une astuce consiste à écrire @ dans l'accessoire de bureau NotePad et à faire un copier-coller.

*[NDLR: Les problèmes de caractères spéciaux sous TML Pascal viennent en fait de l'outil TexEdit de GS/OS. Tout cela a été résolu de façon élégante par l'accessoire "**Stricto Sensu GS**" et l'ensemble de patches à GS/OS publiés par Yvan Kœnig dans Pom's 48. Nous les recommandons chaudement à tous les utilisateurs du TML.Voyez aussi le dossier DERNIERE HEURE de la disquette de ce numéro.]*

# Made in France (●)

Bernard Fournier

## Fontrix GS 6.4

Avant de produire les génialissimes Nucleus et Modulæ (présentés dans ToolBox Mag 1), les joyeux lurons de Dijon avaient réalisé ce logiciel d'écriture sur page graphique. Entièrement compatible avec les fontes Fontrix (bien connues des anciens de l'Apple //), ce logiciel dispose également de plusieurs commandes de style: détourage, coloriage, grossissement, inversion, décalque... et bien d'autres! Vous chargez votre image graphique (format usuel PNT, SCR ou APF) et vous écrivez avec vos plus belles fontes un message d'amour à votre bien-aimé(e). Des centaines de fontes récupérables, des possibilités sans fin... un logiciel indispensable pour les possesseurs des disques de fontes Fontrix. Et bien sûr, fidèles à leur devise: tout cela *sans outils*! et de surcroit gratuit!

*F.T.A., 34 rue des Rotondes, 21000 DIJON*

## Jeu de l'oie 1.0 et Logimots 1.2

Mlle Carole Fox perpétue l'esprit Apple II GS dans l'Océan Indien. Ces deux logiciels écrits en TML Pascal démontrent sa maîtrise de la programmation. 'Jeu de l'oie' est la réplique du classique jeu qui enchanta notre jeunesse: la gestion des dés et des déplacements est confiée au GS, et les moments forts du jeu sont accompagnés d'effets sonores. Très simple d'emploi, convient parfaitement aux petits.
'Logimots': une sorte de Master Mind, mais avec des lettres. Le but du jeu: découvrir un mot de la langue française en suivant les règles du Master Mind. Plusieurs niveaux de difficulté (mots faciles ou rares), aide possible grâce à une énigme expliquant la signification du mot secret...
Ces deux logiciels sont accompagnés, à la demande, de leurs sources TML. Une initiative trop rare et qui honore Mlle Fox, d'autant que ses productions sont gratuites.

*Mlle Carole Fox, Logement de fonction Ecole Candide Azema, 1 rue Mgr Mondon, 97400 Saint Denis, LA REUNION*

## Softword 2.0 et Jeu de Lettres

Des logiciels pour amateurs de Scrabble et autres jeux de lettres.
'Softword' est un Scrabble sur GS avec un dictionnaire de 80 000 mots. La dernière version (2.0) disponible depuis le 15 Octobre est parfaitement au point: réflexion rapide (même sans Transwarp !), possibilité d'aide (l'ordinateur vous indique le meilleur mot... mais à vous de trouver où le placer !), graphisme et effets sonores soignés... une qualité 'pro' pour la modique somme de 100 F. Un seul défaut: on ne peut pas résoudre de problèmes... mais la version 3.0 est en chantier !
'Jeu de lettres': écrit en Basic en 88, ce logiciel est constitué de 3 jeux. *Les anagrammes*: intervertir les lettres d'un mot pour en constituer un nouveau; *Buggle*: faire des listes de mots avc un groupe de lettres, et enfin *Mot le plus long*: écire le mot le plus long avec des lettres données.

*Jean Pierre Grandemange, 27 rue Leopold Bourg, 88000 EPINAL*

## Cluedo 2.0

Après quelques NDA pour se faire la main (NDA chargeur, Formatteur et Prodos NDA), Mr Wolkowitch s'est attelé à la transposition sur GS du célèbre jeu de société. Résolvez l'intrigue policière avec l'aide du GS: découvrez le criminel grâce aux indices qui parsèment le jeu. Si la réalisation du logiciel est parfaite, ce jeu souffre d'un défaut rédhibitoire: l'ambiance chaude des parties sur table manque cruellement...
Pour tout paiement, l'auteur demande juste une carte postale.

*Francis Wolkowitch, 4 bis rue de Lyon, 75012 Paris*

## Mots pour petits, Multiplication, Sakoz

De l'utilisation d'HyperStudio à des fins pédagogiques. Alphabet et Multiplication sont des piles HyperStudio. Il faut résoudre des multiplications ou écrire un mot (on peut même écouter les lettres de l'alphabet). Idéal pour les enfants découvrant l'écriture et le calcul.
'Sakoz' est la francisation des outils sonores 51, 52 et 53: il vous est désormais possible de faire parler le GS en Français en programmant en C. Juste une question: la version en Breton est elle prévue?
Ces logiciels sont gratuits... mais une petite carte postale ferait grand plaisir à l'auteur.

*Jacques Liautard, 13 avenue de la Libération, 95240 Cormeilles en Parisis*

## Miniprix 0.2

La chaude ambiance des circuits de F1, le crissement des pneus, la joie de la pole position, et la récompense suprême: être en tête du championnat !
Ce jeu est d'un réalisme époustouflant. L'auteur a su éviter les pièges habituels: pilotage aisé, pas de carambolage 'mortel' (quelle que soit votre dextérité, vous finirez toujours le circuit).
En page de présentation l'auteur avoue son goût immodéré pour la moto... hélas, il cherche des sponsors pour s'offrir l'engin de ses rêves.... alors, un beau geste: envoyez lui cinquante francs, vous ne le regretterez pas: ce jeu est vraiment super!

*Dany Bär, 1 bd de Chantemerle, 73100 Aix les Bains*

*Comment figurer dans cette rubrique? Il faut avoir écrit une application, un accessoire de bureau, un CDEV, une INIT ou tout autre 'code' fonctionnant sur GS et accepter que ses noms et adresses soient publiés dans Toolbox-Mag, afin que les lecteurs intéressés puissent se procurer le logiciel directement auprès de l'auteur.*

*Bien entendu, il convient de faire parvenir à Toolbox-Mag un exemplaire du programme en question, afin que nous puissions en faire un descriptif aussi complet que possible. Préciser si le programme est distribué en freeware ou en shareware.*

*Les lecteurs demandant à un des auteurs de leur faire parvenir un exemplaire freeware de sa production seront sympathiques de penser que 'gratuit' ne signifie pas 'sans frais'... alors une enveloppe timbrée à 3,80 F accompagnant une disquette pour le retour semble être le minimum.*

# Bidouilles, Patches, Trucs... (🍎)

J.Y. Bourdin

## Quitter proprement le Finder

Que ce soit sur GS ou sur Mac, le Finder est une application entre autres, parfaitement facultative pour l'utilisateur, même si Apple veut nous faire croire le contraire, et même si beaucoup d'utilisateurs croient Apple.
Sur GS, Apple est quand même plus franc: le Finder est muni d'une option Quit qui permet de retourner à l'application qui a lancé le Finder. Apple conçoit donc officiellement que le Finder ne soit pas le lanceur obligatoire, et que les connaisseurs préfèrent Prosel, TransProg ou Wings.
Malheureusement, cette option "Quitter normalement comme tout le monde" n'est pas celle que le Finder propose par défaut. Il faut donc cliquer la souris en plus de faire 🍎-Q et Return: scandaleux, n'est-ce pas?
Voici donc un patch (pour le Finder du système 5.02 français, le fichier appelé /SYSTEM/START sur les disquettes système Apple), qui permet que le Finder propose l'option du Quit normal par défaut. Cherchez la chaîne:
EE 6C 7E 60 F4 00 00 F4 00 00 A2 0B 22,
et remplacez-la par:
EE 6C 7E 60 F4 00 00 F4 02 00 A2 0B 22.
C'est tout.

## Surprise

Donnez à votre fichier /SYSTEM /EXPRESSLOAD un type auxiliaire de $5254 (5452 si vous faîtes ça à l'éditeur de blocs), et rebootez. Surprise...

## Verboten

Supposons que vouliez mettre le tout récent Prodos 8 version 1.9 sur votre GS/OS 5.02 français: ce n'est pas possible, parce que le 5.02 a été conçu pour marcher avec Prodos 8 version 1.8, et pas avec le nouveau. Et même si c'était possible, il ne faudrait pas le faire: c'est **strictement interdit** par le Bureau Politique.
Donc, supposons que vous localisiez dans le fichier START.GS.OS de votre catalogue /SYSTEM la chaîne hexa:
C9 7D 3D D0 0C AF FA 51 00 29 FF 00 C9 02
Supposons aussi que vous la remplaciez par:
C9 7D 40 D0 0C AF FA 51 00 29 FF 00 C9 03
Supposons enfin que vous mettiez le Prodos 8 1.9 sous le nom de "P8" dans votre catalogue /SYSTEM, et que vous lanciez le tout, histoire de voir si ça marche.
Eh bien, soyez prévenus: si vous vous faîtes prendre, **Toolbox-Mag vous désavouera**, en public au moins!

## GS Souffrant

Quand vous mettez sous le capot une TransWarp 7Mhz, une SCSI DMA, plus pas mal d'autres choses, vous portez votre GS à ses limites: il souffre, donc il peut arriver qu'il plante. J'ai essayé ce mélange détonant sur trois GS Rom 01 différents: ils ont souffert (on m'a dit que le Rom 03 était plus résistant).
Ca s'est arrangé en passant par la commande "DisableDataCache" de la TransWarp. Si votre GS accéléré souffre, essayez: cela se résume à *JSL BCFF4C*. J'ai été obligé de caser cet appel très près du départ, dans le fichier appelé "Prodos", sans quoi ça plantait pendant le boot.
Faîtes-nous savoir si vous désirez un patch plus élaboré, ou si "casez quelque part JSL BCFF4C" vous suffit...

## Patch wanted

Je n'ai pas le temps de le faire, mais ça me serait utile, et pas seulement à moi: franciser les caractères correspondant à nos caractères en mode texte dans la fonte interne utilisée, pour ses affichages, par l'excellent Prizm d'Orca (la version TextEdit de l'Editeur de Prizm n'ayant pas l'air d'être pour demain).

Je n'ai pas approfondi, mais si vous cherchez, dans le fichier ORCA du sous-catalogue /PRIZM.MODULES, la première occurrence de la chaîne 3F F0 F0 0C, j'ai bien l'impression qu'il commence là quelque chose qui ressemble fort à une BitMap...
Cette fonte n'est très probablement pas une fonte QuickDraw, l'affichage va trop vite. Mais elle est vraiment très lisible et pratique pour les affichages écran: tant que vous y êtes, si vous en faisiez une fonte QuickDraw...

## Un secret d'AppleWorks

Un petit secret d'AppleWorks-GS (*merci, Mr Bonnet*), pas écrit dans la documentation, pour doubler automatiquement la taille des images pour imprimer en condensé (*cf Toolbox-Mag 1, p.11*).
Mettez votre graphique dans un fichier graphique AppleWorks. Sélectionnez avec l'icône du carré en pointillés la partie d'image que vous voulez doubler en hauteur. Copiez-la dans le presse-papier.
Faîtes ensuite un fichier PAO, et choisissez Condensé dans le menu Page Setup. Collez votre morceau d'image depuis le presse-papier dans votre page PAO. Cliquez sur ce morceau d'image pour le sélectionner. Descendez tout en bas à droite du dessin: deux poignées sont visibles, l'une à gauche, l'autre à droite. Placez le curseur sur la poignée de droite, appuyez sur la touche Shift, et descendez la poignée d'un demi-millimètre. Hop, votre dessin est automatiquement doublé en hauteur!
Vous pouvez bien entendu repasser le tout dans un fichier graphique pour le retravailler, le quadrupler, etc.

## Prosel: macros cachées

Vous le savez sans doute, il existe une mini-sorte de macros dans Prosel: vous pouvez appeler un des modules de Prosel et lui transmettre des commandes, dans un écran Prosel, par la rubrique 'Startup'. Ce qui n'est pas clairement dit, c'est que le caractère ';' n'est pas seulement un "séparateur" de ces commandes: très exactement, *c'est l'équivalent d'un Return au clavier*.
Supposons par exemple que vous désiriez nettoyer automatiquement tous les 'back-up bits' de votre disque dur. Vous faîtes alors l'écran suivant:

*Title Screen*: Clear back-up Bits.
*Prefix*: ?
*Application Path*: B [équivalent du 🍎-B qui appelle le module de Back-up]
*Startup*: C;HD1 [équivalent du C de 'Clear Backup bits' et du nom du dur].

Sauvez cet écran, et lancez votre rubrique de nettoyage: Prosel va lancer le processus, mais s'arrêtera en attendant votre Return. Il s'arrêtera une seconde fois après le nettoyage, attendant un second Return. Pour éviter cela, il suffit de rajouter deux ';' après le nom du dur.
*Startup*: C;HD1;;
Si l'écran de Startup était plus long, on pourrait faire de sacrées macros avec ce système...

# GS News (🍎)

Eric Weyland

*Ce qu'il faut savoir*

## Lettre ouverte de John Sculley

Le PDG d'Apple USA, John Sculley, a fait parvenir à la presse spécialisée une lettre ouverte dans laquelle il explique la position d'Apple relativement au marché de l'Apple II.
Sculley commence par assurer aux utilisateurs qu'Apple continuera la ligne Apple II pour les années à venir. Il déclare, ensuite, que les utilisateurs d'Apple II sont des passionnés, très attachés à la marque Apple, et que leurs appréciations sont toujours encourageantes pour les ingénieurs d'Apple. Sculley reconnaît que c'est l'Apple II qui a fait le succès d'Apple.
Il nous fait ensuite la liste des produits Apple II qui sont sortis durant les deux dernières années: Système 5.0, Apple II Video Overlay Card, Apple II High Speed SCSI Card.
Sculley insiste sur le fait que le GS et le Mac sont des ordinateurs complémentaires qui peuvent partager les mêmes périphériques et les mêmes données; la force de ces machines étant de s'intégrer sans aucun problèmes dans un réseau AppleTalk. Il annonce l'arrivée de nouveaux Macintosh dont un modèle au moins sera doté de l'émulation Apple II... Une traduction de cette lettre a été publiée dans la quatrième édition du Guide de l'Apple II (voir plus bas).

## Apple Expo / Mac II Expo ?

Le CNIT, c'était vraiment l'horreur: une chaleur épouvantable et surtout que du Mac II à perte de vue; partout des costumes trois pièces, et des pantins savants articulant une démonstration mécanique du dernier logiciel de CAO de la société "machin".
Quelque part sur l'expo, une fausse note dans ce concert. Sur le stand ToolBox, pas de crétins-cravates, pas d'esclaves de l'informatique, mais seulement des GS et des utilisateurs sachant en tirer tout le suc. Parfois une question du visiteur du dimanche 'c'est quoi ce petit Mac couleur?' Mais c'est un GS ! lui répond le public de connaisseurs attiré par les prouesses musicales et graphiques de leur ordinateur chéri.
Tous les amoureux du GS sont passés sur le stand: les anonymes ont surtout regardé les dernières créations et démonstrations des programmeurs de talent. Ils sont tous venus, ces programmeurs, montrer ce que l'on pouvait faire avec un GS! Un badaud est passé et nous a dit: "moi je ne travaille que sur Bull Micral, je ne connais pas ToolBox, mais ça a l'air plus sympa que l'ambiance du Forum PC..."
Des programmes ont été présentés sur le stand ToolBox en avant première: Rastan de Taito bien sûr, Hostage de Mindscape, Synthlab d'Apple Computer. (Pourquoi des démonstrations de ce genre n'étaient-elles pas possibles sur le stand d'Apple? Eh, personne n'y aurait cru...)
Il n'était pas rare de voir les programmeurs mettre une dernière touche à leur programme avant de le présenter, en changeant quelques menus octets, et en ré-assemblant le tout sous les yeux ébahis des néophytes. Il était amusant de voir Olivier Goguel, l'auteur de Photonix II, vaillant défenseur de la doctrine 'NO TOOLS', demander à Jean-Yves Bourdin une disquette système GS/OS bourrée de Tools afin de lancer Bouncing Bluster de Jean François Doué et Jean Michel Vallat.
François Uhrich était là. En attendant la belote, il a présenté son jeu de Tarot et souvent le GS lui même au public attentif. Les auteurs de Space Shark, Claude Pélisson, Pascal Watel et Philipp Olivier nous ont montré à quel point leur jeu était rapide et bruyant.
De nombreux contacts ont été pris, de nouveaux projets vont aboutir... Tenez-vous au courant en lisant ToolBox Mag.

## GS/OS: Système 5.03 (Apple Computer)

Une note technique récente d'Apple mentionne les caractéristiques du système GS/OS 5.03. Ce système est disponible par l'intermédiaire de

### Sens Unique

L'encadré d'Eric Weyland, dans notre numéro 1, concernant un programme permettant d'installer des virus sur les disques GS, a donné un coup de pied dans une mini-fourmilière. Cela nous donne l'occasion d'une mise au point.
Il va bien falloir que chacun se fasse une raison: il existe désormais en France un magazine qui prend au sérieux et qui soutient l'Apple II GS. Entre autres choses, cela signifie qu'il n'est plus possible, comme c'était le cas auparavant, de dire ou faire n'importe quoi sur le GS sans que cela tire à conséquence. Désormais, on prend Toolbox-Mag sur le dos, qu'on soit un petit bidouilleur de Landernau ou qu'on soit Apple Computer.
Un programme qui permet à celui qui le veut de mettre des virus sur des disques GS est une arme contre le GS. Il se fait donc épingler par Toolbox-Mag, qui fait ainsi son travail, et qui continuera à le faire, que cela dérange ou non.
Précisons bien: il n'y aura dans les colonnes de Toolbox-Mag aucun dialogue, aucun débat entre les partisans du GS et ses adversaires, quels qu'ils soient. **Toolbox-Mag est à sens unique,** il est tout entier d'un seul côté: celui du GS.
L'autre côté a déjà les colonnes de la "presse dévote". Il faudra qu'il s'en contente.

J.Y. Bourdin

l'APDA. Ce n'est pas une version majeure, mais c'est un événement important: *le 5.02 contenait encore beaucoup de bugs, dont certains très graves* (bugs du SCSI, de TextEdit, du Resource Manager, du Standard File, bugs de GS/OS, etc). Et cette nouvelle version amène nombre de bonnes surprises...

Sur la disquette System.Disk, les fichiers suivants ont été mis à jour: Console.Driver (driver pour le clavier et l'écran), Printer, Modem (drivers de gestion du port imprimante et du port modem).

Dans le sous-catalogue /System, les fichiers Error.Msg (messages d'erreur de GS/OS) et ExpressLoad ont été modifiés. Les FST (File System Translator) Char.FST et Pro.FST ont évolué. La version de ProDos 8 est maintenant le ProDos 1.9. Tous les fichiers GS/OS ont été modifiés (GS.OS, GS.OS.Dev, Start.GS.OS). Le Resource Manager en est maintenant à la version 1.1. Les fichiers TS2 et TS3 (patches pour les outils se trouvant en Rom) ont été modifiés.

Le sous-catalogue /System/Tools contenant les outils chargeables en Ram connaît un certain nombre d'évolutions; voici les *nouvelles versions des outils disponibles avec le système 5.03:*

Tool014 : Window Manager version 3.2
Tool015 : Menu Manager version 3.2
Tool018 : Quick Draw Auxiliary version 3.2
Tool020 : Line Edit version 3.2
Tool021 : Dialog Manager version 3.3
Tool023 : Standard File version 3.2
Tool027 : Font Manager version 3.2
Tool028 : List Manager version 3.2
Tool034 : Text Edit version 1.2

Les fichiers de définitions du Finder (Ftype.Main et Ftype.Aux) ont été modifiés. Basic.System en est à la version 1.4.1, qui corrige une erreur dans la gestion du marqueur EOF (End Of File).

Dans la nouvelle version de ProDos 8, deux bugs ont été supprimés: un problème lié au chargement des programmes d'une taille supérieure à 38 Ko, et un problème qui survenait lors de l'affichage du message d'erreur "Program too large".

ProDos 8 a désormais une routine de Quit spéciale pour les Apple II 65C02 munis d'une carte 80 colonnes. A la place des messages 'ENTER PREFIX' et 'PATHNAME', un mini sélecteur d'applications est maintenant en place. Bravo Apple pour ces sept ans de retard...

La disquette System.Tools a subi quelques changements: les CDevs ATIWriter (ImageWriter), ATLQwriter (ImageWriter LQ) et ATLWriter (imprimante Laser) ont évolué. Le NDA VideoMix.NDA (gestion de la carte Overlay) est modifié. Le FST Appleshare a évolué.

Deux nouveaux drivers, et c'est une très bonne surprise font leur apparition: le très attendu **driver d'imprimante ImageWriter LQ** (un véritable driver exploitant les capacités de cette imprimante) et la présence surprenante d'un driver pour l'imprimante *Deskjet Plus* de chez Hewlett-Packard (imprimante à jet d'encre). Tous les autres drivers ont été modifiés: ATalk, ImageWriter, Modem, Printer, SCSI.Manager, SCSICD.Driver, SCSIHD.Driver, UniDisk.3.5.

En ce qui concerne la disponibilité de la version française du 5.03, voyez l'Apple II Service Team d'Apple France: *c'est son travail*. Quand vous avez acheté votre GS, vous avez payé pour ce service (qui n'a pas à être gratuit, mais qui doit exister).

## Retard GFP: patience...

Le logiciel Gestion Familiale Personnelle (GFP) n'était pas disponible sur l'expo. Nous nous sommes un peu emballés dans le numéro 1 de ToolBox Mag. Deux raisons expliquent ce retard.

D'abord, l'auteur (qui veut pour le moment garder l'anonymat) n'est pas un programmeur à plein temps, et comme beaucoup d'entre nous il doit passer le plus clair de son temps, au bureau, à travailler sur des ordinateurs beaucoup moins amusants que l'Apple II GS. Ensuite, GFP n'a pas encore été véritablement testé, il renferme donc des bugs très gênants lorsqu'il s'agit d'une gestion de compte en banque... Nous avons donc préféré différer sa sortie afin de vous offrir un GFP 'bug free'; le Père Noël devrait vous l'apporter dans sa hotte...

## Roms 01: Mise à jour gratuite, c'est terminé

Votre Apple II GS n'a pas encore les Roms 01 et le nouveau VGC? Tans pis pour vous. La mise à jour coûte désormais le prix de l'échange de la carte mère. Après deux ans de gratuité c'est un peu normal... Et ne rêvez pas ! vous n'aurez pas pour autant une carte mère GS avec des Roms 03. Pour cela il faut acheter une unité centrale complète.

## Soft Sonic Blaster (Applied Engineering)

Le logiciel fourni avec la carte stéréo et de digitalisation Sonic Blaster en est à la version 1.1.

## Photonix II version 2.10 (Toolbox)

Photonix II, le super-copieur FTA, en est à la version 2.10: compatibilité AppleTalk, option Copie en masse pour deux lecteurs, lancement sous Finder sans passer par P8. Mise à jour gratuite.

Photonix II n'est toujours pas protégé contre la copie, mais reste protégé contre le piratage...

## Bouncing Bluster (Toolbox)

Les premiers exemplaires de Bouncing Bluster, c'est-à-dire ceux vendus pendant les deux premiers jours d'Apple Expo, ne fonctionnent pas avec moins de 1.3 méga de Ram. Les acheteurs enregistrés recevront la mise à jour, qui tourne avec 1.2 méga.

## ProSel 16 (Glen Bredon)

La dernière version de ProSel 16 renferme deux nouvelles caractéristiques. D'abord, un accessoire de type CDA (Classic Desk Accessory), appelé Appointments est installable dans le sous-catalogue System/Desk.accs de votre volume de boot. Il s'agit d'un accessoire simulant la tenue d'un carnet de rendez-vous avec la possibilité de programmer une alarme à l'approche de l'heure H. L'autre innovation consiste dans la possibilité d'appeler une calculette scientifique à partir de ProSel 16.

### TML = Complete Technology

La société TML a vendu les droits de ses produits Apple II GS (Tml Pascal, Tml Basic, Tml Source Code Library) à la société Complete Technology. Cette dernière est présidée par Vince Cooper, l'ancien responsable produits de TML Inc. Les produits TML ont maintenant pour noms: Complete Pascal, Complete Basic et Complete Source Code Library. Complete Technology, Inc - 5411 Ortega Blvd. - Suite 7 - Jacksonville, Fla. 32210, USA.

### Checkmate = Micro Memory

Checkmate refait surface sous un autre nom: Micro Memory. La nouvelle adresse est: 7655 East Gelding, £B1, Scottsdale, Ariz. 85260, USA. L'ancien Président de Checkmate, Earl North, reste en place chez Micro Memory.

### Adieu Pom's

Pom's a envoyé un courrier à ses abonnés: après avoir arrêté sa parution en kiosque au numéro 50, elle cesse définitivement sa publication.
Depuis Septembre 1981, Pom's nous a informés et formés à l'utilisation et à la programmation de tous les Apple. Elle a lancé et maintenu le principe d'une revue indépendante, écrite essentiellement par ses lecteurs, ce qui fut une originalité décisive dans la presse informatique mondiale.
Nombre de collaborateurs de Toolbox-Mag ont fait leurs premières armes à Pom's. Certes, Toolbox-Mag, magazine indépendant écrit par ses lecteurs, reprend le flambeau: mais pour le GS seulement. Pour le Mac et pour l'Apple II 8 bits, Pom's va laisser un vide.

### Adieu Ingenuity

La société américaine Ingenuity qui a lancé sur le marché de nombreux périphériques pour Apple II (Disque dur interne Inner Drive, carte d'extension mémoire GS Juice +, disque dur interne pour Apple IIc) cesse toute activité. Pour le moment, nous n'avons pas d'information concernant le support technique des produits diffusés par Ingenuity.

### Adieu Buyer's Guide

Le groupe de presse Redgate Communications va abandonner la publication de la revue The Apple II GS Buyer's Guide après le numéro d'automne 1990. Le magazine Incider a racheté la liste des abonnés qui deviennent de fait abonnés à A+ / Incider.

## *Le Hard*

### TouchWindow (Edmark)

Il existe maintenant une version ADB (Apple Desktop Bus) de la TouchWindow; ce périphérique, qu'on peut chaîner au clavier du GS, permet de contrôler un programme GS en déplaçant le curseur par le mouvement d'un doigt sur la surface de l'écran, au lieu de la souris.

### Zip Chip GS (Zip Technologies)

La carte accélératrice Zip Chip 8000 portant la vitesse de l'Apple II GS à 8 Mhz est maintenant disponible. 16 vitesses intermédiaires sont sélectionnables, la carte est compatible DMA (Direct Memory Access), elle traite d'une manière indépendante le haut-parleur, le joystick et la vitesse des slots, elle a une mémoire cache de 8 Ko étendable à 64 Ko. Elle est bien entendu compatible avec tous les logiciels et matériels existants sur Apple II GS.

### PC Transporter (Applied Engineering)

Le logiciel système pour la carte PC Transporter est maintenant capable de gérer bien plus convenablement les claviers étendus et les disques durs. Il est possible d'installer deux partitions MS/Dos-beurk de 32 Mégas chacune sur un disque dur formaté sous ProDos. Cet upgrade est payant aux U.S.A., sauf si vous avez acheté un lecteur 3.5" Æ, un disque dur, ou un clavier style IBM Æ.

### Alimentation 60 watts (Applied Engineering)

Voici un bloc alimentation plus robuste (60 watts) destiné à l'Apple II GS (l'alimentation standard du II GS est de 38 watts). Cette alimentation peut, au moyen d'un switch, se commuter en 220 volts - 50 hertz. Pratique pour les GS gonflés à bloc et en manque d'électrons.

### RamFast / SCSI (CV Technologies)

Enfin une carte SCSI rapide... La carte RamFast / SCSI est la carte d'interface SCSI la plus rapide disponible sur Apple II GS. Avec elle, il faut 6 secondes pour se retrouver sous Finder au moment du boot, et 5 secondes pour lancer AppleWorks GS. C'est plus de deux fois plus rapide que ce que l'on obtient avec la carte SCSI High Speed d'Apple. La RamFast / SCSI peut gérer jusqu'à 8 partitions. La carte est construite autour d'un micro-processeur cadencé à 10 Mhz et dispose de 256 Ko de mémoire cache. Le taux de transfert des données est de 1 méga par seconde sur Apple II GS et de 980 Ko par seconde sur Apple IIe. Bien entendu, la carte est compatible DMA.

## *Le soft: les jeux*

### Rastan (Taito)

Le petit dernier de chez Taito (Arkanoid 1 et 2, Qix, et bientôt Operation Wolf) est vraiment d'une qualité rare. Il a été programmé par John Brooks, le programmeur de Tomahawk et de Hunt for the Red October.
Vous êtes le héros, une sorte de barbare musclé, disons pour simplifier un Conan le destructeur. Vous devez libérer le royaume d'un dragon maléfique. Le combat va être long et difficile. Il y a de nombreux ennemis à abattre: guerriers, démons ailés, squelettes, monstres, chauves-souris, poissons volants, chevaliers, sorciers, magiciens, anges de la mort... Pour cela vous devez faire attention tout au long du parcours à ramasser les armes (l'épée de feu est la plus efficace), potions et objets divers qui vont faciliter votre combat et préserver vos points de vie.
Les animations graphiques sur plusieurs plans sont spectaculaires, le décor général change à chaque passage de niveau, et à chaque fois, c'est un véritable émerveillement: paysage de

désert, château-fort, forêt équatoriale, forges de Satan. Bien entendu la musique qui accompagne cet ensemble est à la hauteur de l'environnement. Nous vous conseillons donc de vous doter d'une carte stéréo dans les délais les plus brefs...

Attention, vous aurez de nombreuses embûches à déjouer: chausse-trappes, piques, flèches, laves en fusion, chutes de pierres, murs à bascule, traversées de fleuves... Le combat final contre le dragon est très délicat, il vous faudra rassembler tout ce qui vous reste de force en vous pour terrasser l'horrible créature.

Après ce combat vous assisterez à la scène finale... Dommage ! Aucune belle princesse captive à délivrer, pas de vestale soumise vous sautant au cou... Le romantisme se perd aussi dans le jeu d'arcade; "l'Amour n'est vraiment plus un rapace qui plane au dessus de nos têtes..." .

C'est quand même le jeu le plus étonnant que l'on ait vu sur Apple II GS; la démo de Sword of Sodan est du même genre, mais Rastan est un jeu terminé... Bravo encore à John Brooks pour cet exploit de programmation.

## Hostage (MindScape)

Jeu d'arcade dans lequel il faut libérer une ambassade envahie par une horde de terroristes. Attention aux tireurs d'élite, et pensez avant tout à protéger la vie des otages. Parcours sous le feu ennemi, largage d'hélicoptère, tir au fusil à lunette, descente en rappel, rien de vous sera épargné avant de pouvoir donner l'assaut final. Très bon niveau graphique et sonore. Un jeu qui n'a rien à envier à la réalité.

## The Immortal (Electronic Arts)

Encore un jeu d'arcade-aventure qui se passe dans une forteresse médiévale. Oui, mais quel jeu ! Il est programmé par Will Harvey, l'auteur de Zany Golf, cela vous donne déjà une idée de la qualité des graphismes et du son de cette superbe recherche.

Votre maître a été emprisonné dans un donjon, au fin fond d'une forteresse, à vous de la retrouver et de le délivrer. Ramassez les pièces d'or, les anneaux magiques, les sortilèges, les armes, les boules de feu. Tout cela vous sera utile quand il vous faudra combattre les créatures diaboliques qui peuplent ce labyrinthe. Une fois la victoire acquise, n'oubliez pas de fouiller le cadavre de votre adversaire. Faites un plan, et surtout attention aux pièges qui ne vous épargneront pas...

## Pipe Dream (Lucas Film)

Poser des canalisations, cela semble facile. Avez vous déjà pensé aux différentes formes de tubes: droits, coudés à gauche, coudés à droite, coudés vers le haut, coudés vers le bas, tubes en té... La plomberie c'est un métier ! Les tubes doivent être posés proprement sans enchevêtrement superflu. Attention à la cadence: un mauvais plaisant a eu la bonne idée de rebrancher l'eau; elle s'engouffre dans votre canalisation; attention aux fuites et aux circuits non fermés...

Plus l'eau fait de chemin, plus vous marquerez de points. Les tubes se posent sans fer à souder, à la souris, au joystick ou au clavier. Vitesse et anticipation feront de vous un champion de Pipe Dream. Les tubes bien placés font gagner des points; tout matériel gâché vous en fera perdre.

J'ai oublié de vous dire qu'il n'est pas question de placer les tubes de son choix (ce serait trop facile); le type de tube à poser est aléatoire; pour vous aider, vous connaissez à l'avance les cinq tubes que vous allez pouvoir poser. Il existe plusieurs niveaux de jeu; un système de mot de passe permet de passer d'un niveau à l'autre. A chaque changement de niveau, on peut gagner des points en empilant des tubes, pour former une canalisation, dans un jeu de genre Tetris. Les niveaux supérieurs sont réservés à l'élite de la plomberie; attention, des obstacles vont venir contrarier vos projets de modernisation.

## Space Ace (Ready Soft)

Ce jeu, dont je n'ai vu pour le moment qu'une démo, fait immédiatement penser à l'ambiance d'une bande dessinée: super-héros, navettes spatiales, fusées, soucoupes volantes, rayons laser, paysages lunaires; tout le graphisme de ce jeu est très coloré. Space Ace est un jeu d'arcade qui va vous propulser dans l'univers de la guerre des étoiles. A voir et à écouter...

## Windwalker (Origin)

Superbe jeu d'action à base d'arts martiaux. L'action est inspirée d'un conte de Mœbius. Les décors extrêmes-orientaux vont tout de suite vous plonger dans un univers où le combat, mais aussi la prudence et la ruse, seront nécessaires pour mener à terme votre mission. L'entrainement va être long et difficile, les épreuves variées, avant de pouvoir prétendre devenir "le vrai maître de la défense". Attention, WindWalker n'est pas un jeu de karaté, c'est avant tout un jeu de réflexion...

## Questmaster (Miles Computing)

QuestMaster 1 (The Prism of Heheutotol), est un jeu d'aventures en mode desktop. L'interface utilisateur va faire sourire les nostalgiques des premiers jeux d'aventures de Scott Adams (ceux où il fallait taper Open Door, etc, pour entreprendre une action). Ici, malgré les menus déroulants et la souris, il va vous falloir taper au clavier ce que vous désirez faire... Les mouvements peuvent quand même se faire au joystick; les actions les plus classiques (Inventaire, Regarder...) ont des équivalents clavier. Le jeu en lui-même est intéressant. Vous trouvez dans votre jardin une forme hexagonale verte clignotante; à vous de découvrir de quoi il peut s'agir...

## Cosmocade (Pangea Software)

Brian Greenstone, l'auteur de Xenocide et de nombreux programmes en shareware (Senseless Violence 1 et 2...) continue de nous surprendre en nous offrant Cosmocade; il s'agit en fait de deux jeux d'arcade distincts: Naxos et Calibus. Le premier est un jeu de tir (détruire les monstres); le second vous donne le contrôle d'un vaisseau de l'espace... Comme d'habitude, le son est très travaillé... Brian se plaignant de ne pas recevoir beaucoup d'argent de ses sharewares, je vous donne son adresse: PANGEA SOFTWARE - 10918 KIRWICK - HOUSTON, TX 77024 - U.S.A.

## GS Numerics (Spring Branch Software)

Le programme de mathématiques le plus complet à ce jour sur Apple II GS en est maintenant à la version 1.4. Il est possible d'acquérir ce logiciel et de l'utiliser sur plusieurs postes de travail (dans une école par exemple) en bénéficiant de prix spéciaux.
A l'heure actuelle, GS Numerics est utilisé dans 48 états des Etats-Unis, et 12 pays l'utilisent déjà pour faciliter l'apprentissage de l'algèbre, de la trigonométrie... A noter qu'une version Macintosh vient de sortir et qu'elle dispose de fonctions bien plus avancées; il est vrai que beaucoup d'utilisateurs de Mac ont déjà étendu leur ordinateur à plus d'un méga de mémoire...

## GTV: A Geographic Perspective on American History (National Geographic Society)

Voici l'une des premières applications pour Apple II GS livrée sur compact disk. Il s'agit d'une réalisation de la très réputée National Geographic Society. Le package comprend deux compact disks. Le programme explore d'une manière interactive l'histoire politique et sociale des Etats-Unis.

## Wings (Vitesse)

Sélecteur de programmes destiné à remplacer le Finder ou ProSel, Wings est bien entendu en mode desktop; il permet de conserver 128 chemins d'accès afin de lancer automatiquement l'application correspondante. Il est également possible de visualiser une image graphique.

## CDA Background Sound (Parik Rao)

Ce CDA est à placer d'urgence dans le sous catalogue /System/Desk.accs de votre disque système. Il permet de jouer en tâche de fond, pendant que vous travaillez, des musiques de différents types: les sons digitalisés, les fichiers de musique compressés de type ACE et ASIF, les sons HyperStudio, les musiques SoundSmith, MusicStudio et Synthlab.

## SynthLab (Apple Computer)

Tout le monde sait que la qualité du son du GS provient de la puce Ensoniq. Une grosse tête de chez Ensoniq a été débauchée par Apple USA (il est difficile de faire longtemps la sourde oreille aux propositions alléchantes d'Apple). Mark Cecys, c'est son nom, vient de réaliser SynhtLab qui permet d'utiliser l'Apple II GS comme un séquenceur MIDI et un outil de génération de sons.
La version que nous avons pu essayer est vraiment de grande qualité; nous avons tout particulièrement apprécié l'interprétation du 'vol du bourdon'. Rappelons que toutes ces prouesses musicales sont obtenues par les outils GS.
Précisons tout de suite que SynthLab est un outil sophistiqué et qu'il ne prend vraiment d'intérêt que lorsque l'on possède un instrument de musique MIDI (par exemple un clavier). On dispose alors d'un véritable studio d'enregistrement. SynthLab se pose dès à présent, avec SoundSmith, comme le programme musical de l'Apple II GS.

## PrintShop Companion GS (Broderbund)

Roland Gustafsson a terminé la programmation de PrintShop Companion GS. Avant que le programme ne soit commercialisé, nous pouvons déjà, grâce à nos agents californiens, vous donner en exclusivité, les caractéristiques générales du complément indispensable à PrintShop.
Fabrication couleur de calendriers (hebdomadaires, mensuels, annuels) avec la possibilité d'y insérer des dessins PrintShop; fabrication d'enveloppes personnalisées et d'étiquettes (texte en couleurs, graphiques, bordures...). Un éditeur graphique complet et très rapide vous donnera la possibilité de dessiner vos pages. Le programme comprend aussi un éditeur de bordures, un éditeur de fonds graphiques (dessin automatique et aléatoire de fonds graphiques colorés), et un éditeur de polices de caractères. En prime, vous pouvez vous amuser à élaborer des créatures monstrueuses pour les reprendre dans les autres modules.
PrintShop Companion a le look graphique de PrintShop; l'interface utilisateur y est très étudiée. Avant d'imprimer quoi que ce soit, le programme vous donne la possibilité d'une 'Preview' sur l'écran couleur. Le programme est au standard ProDos. La version qui sera commercialisée devrait contenir une surprise...

## Platinum Paint (Beagle Bros)

Le Mini-Paint de Jem Software est donc finalement devenu un maxi-Paint. Bien programmé, pas trop gourmand en mémoire, Platinum Paint offre quelques fonctionnalités absentes de PaintWorks Gold.

## EuroWorks 3.0 (S.A. AuTeur Co)

Ce programme conçu avec UltraMacros ajoute à AppleWorks 3.0 US une commande d'impression des caractères spéciaux pour le Français, l'Allemand, l'Italien, le Portugais et l'Espagnol. Les symboles spéciaux et les accents sont faciles à obtenir. L'impression est possible sur les imprimantes ImageWriter 1, 2 et LQ. Il est bien entendu possible de sauvegarder les fichiers obtenus par cette technique.
Reste qu'AppleWorks-GS fait tout ça infiniment mieux.

## The New Print Shop 8 (Broderbund)

Une nouvelle version de Print Shop 8 bits est disponible. Une bonne nouvelle, il n'est plus protégé contre la copie, bien que l'auteur n'aie pas tout compris à ProDos. Parmi les nouvelles caractéristiques du programme, nous avons noté: une résolution graphique supérieure, une interface utilisateur plus souple, la possibilité de fabriquer des calendriers...

## Graphic Disk Labeler (Triad Venture)

Graphic Disk Labeler (GDL) permet de donner une touche professionnelle aux étiquettes de vos disquettes. Il est possible d'utiliser les polices de caractères de son choix pour le texte et

de placer des icônes à l'endroit où l'on le désire. Le programme est fourni avec une cinquantaine d'étiquettes qui pourront vous servir de modèles.

### Soft Quickie (Vitesse)

La version 2.02 du logiciel accompagnant le scanner à main Quickie est disponible.

### Drivers

La société Vitesse est sur le point de commercialiser des drivers pour imprimantes Laser Hewlett-Packard. Une imprimante Laser à moins de 10000 Francs, cela donne à réfléchir... Pendant que l'on parle de drivers, nous avons pu tester de nouveaux drivers pour l'ImageWriter II (il ne s'agit donc plus de 'Fumées sans feux'): ce sont certainement ceux qui seront livrés avec le très attendu système 6.0. La vitesse d'impression est plus grande, et sans aucune perte de qualité.

### Desktop File Control (Roger Wagner)

DFC est un petit NDA, paru auparavant en shareware, qui permet d'explorer le contenu de ses différents disques. On peut ainsi savoir rapidement la taille d'un fichier, sa date de création et de modification, son type et son type auxiliaire. DFC peut aussi renommer, déplacer, copier, changer le type et le type auxiliaire, visualiser, accéder aux bits d'accès et effacer tout fichier. Pratique, mais Disk Access (Seven Hills) ou The Desktop Manager (On Three) sont plus complets.

## *Les softs: freewares / sharewares*

### Nifty List 3.0 (David Lyons)

La version 3.0, toujours en shareware, de ce CDA, indispensable aux programmeurs est maintenant disponible. Rappelons que Nifty List est utilisé par tous les fanatiques de programmation sur Apple IIGS. C'est avant tout un super-moniteur, muni d'un désassembleur qui indique clairement (avec leur nom) tous les appels système: appels à la boîte à outils, appels à GS/OS ou à ProDos 8, softswitches, vecteurs... Toutes ces références sont situées dans un fichier séparé: NList.Data; bien entendu, ce fichier est éditable...
La version 3.0 améliore considérablement tout ce qui concerne la manipulation des outils, des handles... Les ressources sont également traitées. Nifty List est donc à jour pour le système 5.0.

### RunQ (Marcel Egloff)

RunQ est un programme en shareware. Il se présente sous la forme d'un PIF (Permanent Initialization File). On y accède en cliquant sur une petite icône tout à fait à droite sur la barre des menus. C'est en fait un launcher accessible à tout moment. A notre avis, en-dessous de TransProg du Maître Uhrich.

### A2FX (Char Wilson)

Apple 2 File Exchange permet de lire directement les disquettes 800k Macintosh (HFS) pour les convertir au format ProDos. On peut tranférer les segments de données, les ressources, ou les deux parties du fichier HFS. Indispensable, c'est l'équivalent de l'Apple File Exchange du Macintosh. Ce logiciel marche parfaitement, et il est en freeware, mais l'auteur accepte les dons!

### Editeur de ressources LLRE (Jason Coleman)

LLRE (Low Level Resource Editor) est un programme américain en shareware. La version 1.1 permet la création et l'édition de tous les types de ressources en hexadécimal ou en ASCII. LLRE donne la possibilité de réparer les fichiers ressources abimés. LLRE dispose d'une caractéristique très intéressante: l'importation d'un segment de données pour le sauvegarder en fichier de ressources; bien entendu, l'opération inverse est possible. Il est aussi possible de supprimer dans un fichier le segment de données ou le segment de ressources. En fait, LLRE permet toutes les manipulations (couper / coller) des ressources d'un fichier à l'autre. Il vous est donc possible de créer des programmes utilisant des ressources communes. LLRE est le premier éditeur de ressources faisant une séparation nette entre données et ressources. Indispensable pour les programmeurs.

## *A lire*

### Le Guide de l'Apple II: 4 ème édition - 2 ème semestre 1990 (Apple Computer)

Comme prévu, il est sorti pour Apple Expo. Comme d'habitude il est disponible gratuitement auprès d'Apple France, et comme de coutume il ne mentionne pas une seule fois l'existence de ToolBox.
Le silence volontaire du Guide sur le Tarot de François Uhrich et sur le livre GS Epluché va-t-il s'étendre aux autres logiciels Toolbox, et à Toolbox-Mag lui-même, toutes réalisations présentées à l'Apple-Expo? Dans cette hypothèse, cela signifierait que le Guide est un outil que se donne Apple-France pour tenter d'étouffer le travail des développeurs français pour le GS...
Le Guide est présenté cette fois-ci dans une édition internationale dont le contenu technique a été rédigé par une équipe de spécialistes. Il contient un résumé de l'histoire d'Apple. La lettre ouverte de John Sculley a été traduite.

### TimeOut Central (A2 Central)

TimeOut Central est une revue sur disquette 800 k traitant d'AppleWorks-8, des macro-commandes et des TimeOut. La revue est classique: courrier des lecteurs, trucs et astuces, TimeOut et Macros en shareware, templates...

### Entendu à l'Expo

"Nous facturons 1200F TTC par poste pour l'installation de la dernière version du système. Bien entendu, le système lui-même est facturé en sus, ainsi que nos déplacements".

*(La prochaine fois que vous cliquerez sur l'Installeur, sachez combien vous gagnez...)*

# Courrier des lecteurs (●)

***Quatre tendances*** semblent dominer dans votre courrier à propos de Toolbox-Mag:
1/ Bravo.
2/ Ne pas oublier les utilisateurs non-programmeurs.
3/ Plutôt de la réticence envers le 8 bits.
4/ Mais une nette faveur pour l'hypermédia.

***Cinq commentaires:***
1/ Merci.
2/ Toolbox-Mag n'est pas fait uniquement ***pour*** les programmeurs: mais il est fait ***par*** eux.
3/ L'expression '8 bits' est ambigüe. Le 8 bits est un mode du 65816. Ce sont les programmes 'compatibles 2E' que vous n'aimez guère.
4/ Une chose est sûre: vous avez un GS! Mais il y a un problème avec l'hypermédia, au moins avec les piles HyperStudio: ça mange de la disquette.
5/ Nous restons à l'écoute.

## Bug de PaintWorks Gold

***Robert Santelli:*** *Composez sous PaintWorks Gold une image comprenant du texte. Imprimez-la sur Imagewriter. Retournez à votre image et choisissez une nouvelle fonte: plantage garanti.*
*Mon remède actuel: sauver le fichier, ouvrir un nouveau dessin, fermer ce nouveau dessin et recharger le précédent fichier. Je peux alors "choosefonter" sans plantage. Il est donc impératif de sauver au moins provisoirement l'écran de travail si l'on ne veut pas tout perdre.*
*Je pense que c'est un bug inadmissible pour un logiciel comme PaintWorks Gold. Qui pourra nous dire les octets à changer avec Block Warden?*

***Toolbox-Mag:*** Merci de signaler ce bug, et votre truc pour le contourner. Activision a cessé tout support pour PaintWorks Gold. Ne reste envisageable qu'un patch d'un utilisateur. Nous faisons donc appel au peuple...

## Quel banc?

***Christophe Quiénot:*** *Après plusieurs essais du patch pour The Graphic Exchange du numéro 1, je n'arrive pas à le faire fonctionner. J'ai un GS Roms 3 avec une GS Ram Plus 4 Mégas. Pourriez-vous me donner une valeur pour la variable A%?*

***Toolbox-Mag:*** Plus besoin de patch. Une version normale de Graphic Exchange, qui marche sous GS/OS, est disponible chez Roger Wagner.

## Précisions

❤ Il s'est trouvé une vingtaine d'entre vous, membres de Toolbox donc abonnés gratuitement aux trois premiers numéros, pour prendre en plus un abonnement pour six numéros sans attendre de rappel. ***Bien entendu, si vous êtes dans ce cas, vous êtes abonné jusqu'au numéro 9*** (même si vous avez reçu deux fois le numéro 1). Et merci de votre fidélité.

⊗ Attention aux habitudes prises sur les serveurs: ***une lettre signée d'un pseudonyme est pour nous une lettre anonyme,*** et elle va directement à la poubelle. Nous avons une définition simple du "lecteur": quelqu'un dont les coordonnées sont dans notre fichier des abonnés.

## Quand on aime...

***Léon Dehertogh:*** *Afin d'apporter une modeste contribution à la diffusion de Toolbox Mag, j'ai photocopié le bulletin d'abonnement en une centaine d'exemplaires, et j'en ai adressé un à tous les amis et connaissances, fanas du GS (17 lettres en tout).*
*Le reliquat a été déposé chez mon concessionnaire Apple, qui va se faire un plaisir de promotionner le magazine. Ce concessionnaire est lui aussi passionné du GS - il a vendu plus de 200 GS - et partant, fort mal noté chez Apple Belgique. Le titre de concessionnaire Apple lui sera d'ailleurs retiré à la fin de l'année.*
*J'ai en outre pris des contacts avec des amis de Liège qui ont promis leur collaboration.*
*J'espère que cela donnera quelques résultats et que le temps consacré n'aura pas été vain. Mais finalement, le temps compte-t-il s'il s'agit de rendre service à des amis et... de défendre le GS?*

***Toolbox-Mag:*** La vraie force du GS n'est pas chez Apple: elle est dans la communauté des utilisateurs. Léon Dehertog est bien un ami, mais ***un ami spécial: nous ne l'avons jamais vu!*** Et nous avons beaucoup d'amis comme lui.
Pour notre autre ami vendeur de GS, qu'il ne se chagrine pas de ne plus être concessionnaire Apple. En France au moins, ce titre fonctionne plutôt comme repoussoir chez les amateurs de GS (et chez les macintoshiens avertis). On peut faire beaucoup pour le GS sans s'occuper d'Apple. La preuve...

## Précisions

***Pascal Pintore:*** *Pourrait-on avoir les adresses de certains des produits US mentionnés dans les GS News?*
*Existe-t-il une documentation française d'Orca/C?*

***Toolbox-Mag:*** L'adresse des éditeurs US est la plupart du temps inutile, car ils ne vendent pas directement. Quant aux revendeurs, vous connaissez les adresses en France.
Si vous achetez Orca/C en France, le vendeur doit, c'est une *obligation légale*, vous fournir une documentation française.

*Revue soft:*

# Block Out (🍎)

**par François Hermellin**

Block Out, de California Dreams, est un jeu de stratégie et de réflexion. Proche de Tetris (il a d'ailleurs été programmé à l'Est!), il reprend l'idée du puzzle basé sur des empilements de cubes, mais innove définitivement en ajoutant l'aspect tridimensionnel.

Des groupes de blocs en 3-D, composés de 1 à 5 cubes, apparaissent à intervalles réguliers au sommet d'un puits formé d'une succession de niveaux. Ils tombent automatiquement dans ce puits peu à peu. Le joueur doit les assembler de telle façon qu'ils occupent tout un niveau du puits, sans laisser d'espace inoccupé entre eux. A chaque fois qu'un niveau est rempli de cette façon, tous les cubes de ce niveau disparaissent, vous laissant plus de place pour maneuvrer. Les groupes de cubes qui tombent peuvent être manipulés dans les trois dimensions pour s'emboîter exactement dans ceux qui étaient déjà présents. Il est possible de les faire pivoter sur tous leurs axes, ainsi que de les déplacer vers n'importe quelle case du niveau pendant leur chute.

Lorsque les niveaux ne sont pas complets, ils s'accumulent. Et quand le bord du puits est atteint, le jeu prend fin. Le défi consiste donc à visualiser le plus rapidement possible la façon optimale de placer les cubes par rapport à ceux qui sont déjà présents au fond du puits, puis d'effectuer les déplacements et les rotations nécessaires.

Différentes options sont proposées. Ainsi il est possible de modifier la taille du puits, le niveau de difficulté, de choisir le mode entraînement ou même Démo. Une aide est également prévue.

Ce jeu est intéressant pour deux raisons principales. Tout d'abord, il est d'un maniement très aisé, voire instinctif. Les blocs répondent parfaitement aux injonctions de la souris. On peut ainsi se consacrer au plaisir du casse-tête en 3-D sans être freiné par des commandes rébarbatives.

Le second point positif réside dans sa formidable capacité à éveiller l'intérêt de l'utilisateur. Le principe du jeu est simple (bien que sa réalisation soit très soignée), mais il arrive à séduire les joueurs les plus blasés. Ces petits cubes ont quelque chose de magique et on se laisse très rapidement prendre au jeu !

## Au boulot, lecteurs !

Un index de tous les sujets abordés dans Toolbox-Mag, ce serait bien pratique, n'est-ce pas? C'est bien ce que nous nous sommes dit, aussi avons-nous décidé de... vous le faire faire!

Sur la disquette de ce numéro, dans le sous-catalogue /INDEX, vous trouverez un fichier base de données Appleworks-GS. Il ne vous reste plus qu'à entrer les données.

Faîtes deux fichiers séparés, pour le N°1 et le N°2. Envoyez-nous les fichiers pleins avant le 20 Décembre. Nous sélectionnerons les deux meilleurs envois.

Toute peine mérite salaire: votre rémunération, si vous êtes sélectionné, consistera en un abonnement gratuit, ou plutôt une prolongation d'abonnement d'un mois par fichier que vous nous enverrez.

Car il faudra également vous engager à nous envoyer l'index de chaque numéro dans un délai d'un mois après sa réception. De cette façon, nous pourrons mettre l'index du numéro 2 sur la disquette du numéro 3, celle du 3 sur le 4, etc.

Et si vous n'êtes pas sélectionné? Rassurez-vous, il y aura d'autres occasions: une idée comme celle-là (faire travailler les lecteurs à notre place), c'est une idée en or!

## Insoumission

- Parmi tous les maudits logiciels qui refusent sciemment de se plier aux règles de l'interface utilisateur Apple, quel est le plus diffusé?
- HyperCard, d'Apple Computer, inclus avec tous les Macintosh, et bientôt (s'il fait beau) avec tous les GS.

JYB

# Toolbox Mag : ça craque !

J.Y. Bourdin

Un 'Directeur de Publication', c'est celui qui dit: "Tu as 50 pages et 800K". Un 'Rédacteur en Chef', c'est celui qui remplit les 50 pages et les 800K. Ca a marché pour le numéro 1, mais, dès le numéro 2, ça ne marche plus. Il y a plus de 50 pages, et plus de 800k. Le Rédacteur en Chef coince, Toolbox-Mag craque!

L'Apple II GS est une magnifique machine, mais précisément il est 'G' (Graphique) et 'S' (Son). Les sons et les graphiques, cela prend de la place. Quant aux programmes, ils sont, et c'est normal, accompagnés de librairies, ressources, units, que sais-je encore: le GS est une machine d'aujourd'hui.

Or le Rédacteur en Chef exclut totalement la solution adoptée par d'autres magazines: demander aux auteurs de réduire leurs programmes. Moins d'images, moins de sons, moins de lignes de programme, ce serait moins de GS. Auteurs de Toolbox-Mag, continuez à prendre vos aises, sans vous occuper de la longueur. Mais ce satané Directeur ne démord pas de ses '50 pages, 800k'. Une seule solution: ***avoir une idée!***

Cette idée, la voici: **la disquette Toolbox-Mag Bis**. Sur cette disquette, nous avons mis tout ce qui n'était pas absolument indispensable aux programmes de Toolbox-Mag 2.

Nous avons mis aussi ce qui, à notre avis, n'intéressera qu'une fraction de nos lecteurs: par exemple, les corrections d'Yvan Kœnig à Orca-Disassembler sont très utiles, mais seulement à ceux qui utilisent Orca-Disasm. Certaines discussions spécialisées et techniques y trouvent aussi leur place.

Ceci dit, le Directeur a raison: vous avez payé pour 50 pages et 800k, et si on vous en donne plus, on coule le magazine. Si vous voulez plus de Toolbox-Mag, il faudra donc le payer en plus. Le prix que nous avons fixé pour cette disquette 'Bis' (100 F port compris) nous permet de ne pas être financièrement perdants, et nous donne la souplesse nécessaire pour ne rien jeter, ne rien perdre de Toolbox-Mag.

Précisons donc bien: la disquette Toolbox-Mag Bis n'est absolument pas indispensable à la lecture de Toolbox-Mag 2, ni à l'utilisation des programmes qu'elle contient. Nous n'avons pas inventé un nouveau produit que nous cherchons à vendre en plus. Simplement, pour ceux d'entre vous qui voudraient 'Plus de Toolbox-Mag', c'est possible.

Notez que la disquette Bis, tout comme la disquette normale, est entièrement sous Copyright Toolbox-Mag.Voici son contenu:

*/ANIM*:
Contient l'animation de présentation de la disquette.

*/COMPLEMENTS.APW*:
Contient des fichiers de compléments de Patrick Desnoues à la dernière version d'APW: un fichier de macros pour l'assembleur, introduisant des commandes bien pratiques comme des branchements longs de ± $8000 octets. Un fichier pour les ressources, introduisant de nouveaux types de ressources non référencées dans le fichier Apple. Explications dans le fichier 'Lisez.Moi'.

*/DISCUSSIONS*:
Vous trouverez dans ce dossier quelques débats à plusieurs voix engendrés par Toolbox-Mag N° 1. A vous, face aux arguments présentés, de vous faire votre opinion.
Faut-il demander à Claris une version française d'Appleworks-GS? Faut-il obéir à la note 53 du Bureau Politique sur le chargement des outils par les NDAs ?

*/GS.PUZZLE.PICS*:
Des images pour GS-Puzzle, choisies sadiquement pour leur difficulté.

*/GS.PUZZLE.SRC.2*:
Contient deux versions imprimables du source du programme GS-Puzzle de Bruno Boissière: PUZZLE1.SA.P est un source sans accents pour imprimer correctement depuis TML Pascal II (qui imprime en mode texte).
PUZZLE1.TEXT est un fichier texte Ascii GS, destiné à être imprimé par des logiciels qui impriment en graphisme, tels Appleworks-GS, BeagleWrite, etc.

*/MERLIN.STARTUP*:
Contient la traduction en Merlin, par Yvan Kœnig, de la routine de StartUpTools de P.Desnoues du numéro 1, ainsi que le source Merlin de l'accessoire. N'oubliez pas de lire le fichier "Lisez.Moi" de ce dossier.

*/ORCA.DISAM*:
Contient des fichiers de data pour le désassembleur d'Orca réalisés par Yvan Kœnig, qui a comblé des manques et rectifié des erreurs. Ces fichiers sont indispensables en particulier pour désassembler correctement des programmes assemblés par Merlin. Lisez le "Lisez.Moi".

*/PICMASTER.GS*:
Compléments au programme PICMASTER.GS de Bernard Fournier: /TML.UNITS contient les versions compilées des Units TML Pascal. /EXEMPLES contient des exemples d'images sous différents formats (y compris les ressources). Ces images ne sont pas seulement intéressantes comme exemples: souvenez-vous de celles du numéro 1.

*/SOUNDSMITH.MUSICS*:
Contient des musiques Soundsmith pour le Tool 219, avec les fichiers d'instruments sauvés dans la forme de wavebank qui convient au Tool219.

# Aidez Toolbox-Mag (*)

Pour que le pari de Toolbox-Mag (un magazine de qualité qui prend l'Apple II GS au sérieux) soit une réussite, une exigence: doubler notre nombre actuel d'abonnés en moins d'un an. Toolbox-Mag, qui n'a pas d'autre ressource que les abonnements, ne peut pas - et ne souhaite pas - s'offrir des encarts publicitaires dans les colonnes d'une presse informatique dite 'générale', mais qui en fait ignore ou méprise le GS.
Il ne peut compter que sur ses lecteurs eux-mêmes comme agents publicitaires: faîtes connaître Toolbox-Mag à vos amis, proposez-leur le bulletin ci-dessous. Un abonnement à Toolbox-Mag, c'est aussi un cadeau de fin d'année qui sera apprécié, soyez-en sûr.
**Vous pouvez aussi nous aider en nous envoyant des adresses d'amis qui ont un GS, mais ne connaissent pas Toolbox-Mag:** nous leur enverrons un courrier. Nous comptons sur vous.

**Bulletin d'abonnement à Toolbox-Mag.** (Le présent bulletin est valable jusqu'au 31/1/91).

*Je m'abonne pour six numéros du magazine Toolbox-Mag (six revues et six disquettes) à partir du numéro 1, au prix de 590F (690F hors CEE). Pour le même prix, je recevrai gratuitement le catalogue Toolbox, et je bénéficierai d'une remise automatique sur tous les produits Toolbox.*

Nom: ..... Prénom: .....
Adresse:.....
Code Postal: ..... Ville: ..... Pays: .....

Ci-joint mon règlement par: ❒ Chèque bancaire ❒ CCP ❒ Mandat
❒ Cochez ici si vous souhaitez recevoir une facture

Le ...../...../....., Signature: .....

## Encore plus de Toolbox-Mag!

**Bon de commande de la disquette Toolbox-Mag N° 2 BIS.**
(Le présent bon de commande est valable jusqu'au 31/1/91).

*Je commande la disquette "Toolbox-Mag 2 BIS", dont le contenu est décrit dans le numéro 2 de Toolbox-Mag, au prix de 100F (130F hors CEE), port compris.*

Nom: ..... Prénom: .....
Adresse: .....
Code Postal: ..... Ville: ..... Pays: .....

Ci-joint mon règlement par: ❒ Chèque bancaire ❒ CCP ❒ Mandat
❒ Cochez ici si vous souhaitez recevoir une facture

Le ...../...../....., Signature: .....

## Je désire faire connaître Toolbox-Mag à mes amis.
## Je vous joins ci-dessous leur adresse, pour que vous les contactiez.

Nom: ..... Prénom: .....
Adresse:.....
Code Postal: ..... Ville: ..... Pays: .....

Nom: ..... Prénom: .....
Adresse: .....
Code Postal: ..... Ville: ..... Pays: .....

**Toolbox-Mag, 6 Rue Henri Barbusse, 95100 Argenteuil**